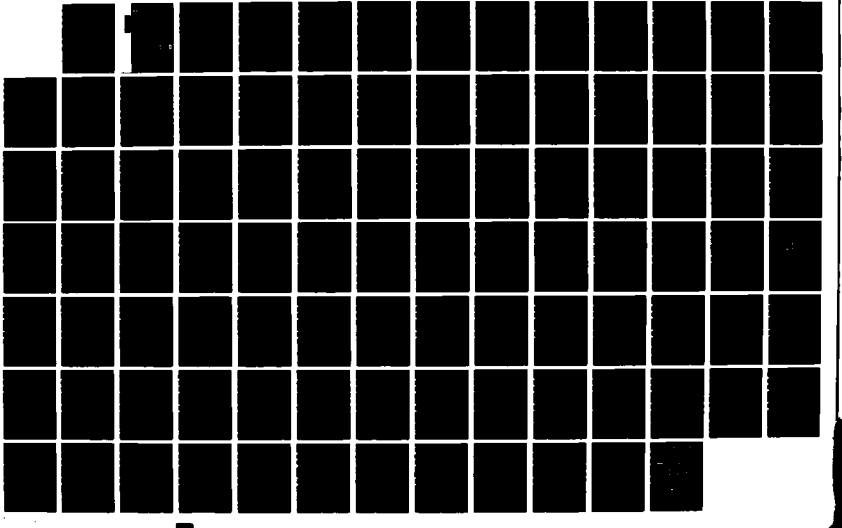
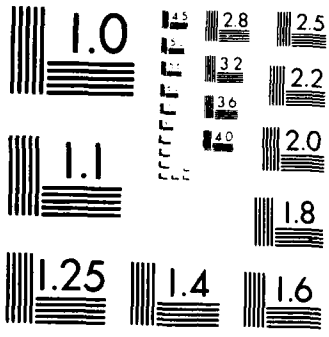
AD-A192 285 MACH2: A TWO-DIMENSIONAL MAGNETOHYDRODYNAMIC SIMULATION CODE FOR COMPLEX (U) MISSION RESEARCH CORP ALBUQUERQUE NM M H FRESE SEP 87 AMRC-R-874 F29601-86-C-0216 1/1

UNCLASSIFIED F/G 20/9 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

AMRC-R-874

Mission Research Corporation

MACH2: A TWO-DIMENSIONAL MAGNETOHYDRODYNAMIC SIMULATION CODE FOR COMPLEX EXPERIMENTAL CONFIGURATIONS

Michael H. Frese

September 1987

Prepared for: AIR FORCE WEAPONS LABORATORY/AWPP
Kirtland Air Force Base
New Mexico 87117

Under Contract: F29601-86-C-0216
Subtask 03-02/00

Prepared by: MISSION RESEARCH CORPORATION
1720 Randolph Road, SE.
Albuquerque, New Mexico 87106

CLEARED FOR PUBLIC RELEASE: AFWL/PA 87-626
22 December 1987

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. AMRC-R-874	2.	3. Recipient's Accession No.
4. Title and Subtitle: MACH2: A Two-Dimensional Magnetohydrodynamic Simulation Code for Complex Experimental Configurations			5. Report Date: September 1987 6.
7. Author(s): Michael H. Frese			8. Performing Organization Rept. No. AMRC-R-874
9. Performing Organization Name and Address: MISSION RESEARCH CORPORATION 1720 Randolph Road, SE. Albuquerque, New Mexico 87106			10. Project/Task/Work Unit No. Subtask 03-02/00 11. Contract(C) or Grant(G) No. (C) F29601-86-C-0216 (G)
12. Sponsoring Organization Name and Address: AIR FORCE WEAPONS LABORATORY/AWPP Kirtland Air Force Base New Mexico 87117			13. Type of Report & Period Covered: Interim 14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

MACH2 is a flexible and powerful two-dimensional magnetohydrodynamic simulation code designed specifically to handle complex experimental configurations. The code's capabilities include a numerically generated solution-adaptive grid which permits it to be used for either Eulerian or Lagrangian flow problems, use of real equations of state and transport properties from the Los Alamos National Laboratory SESAME package, and a multigrid implicit magnetic field diffusion solver which can be used to simulate problems with vacuum.

The descriptive power of the namelist-based problem input language is sufficient that no code modifications are required to set up any of a broad class of problems. Almost no knowledge of the internal data structures and numerical techniques is required to do simulations that realistically model complex experiments. This is made possible by the code's multiblock architecture and table-driven control structure. Code reliability and flexibility are ensured and enhanced by a structured and modular implementation of boundary conditions and block coupling.

This report details the physical model, including boundary conditions; permissible problem geometries; time differencing; and spatial discretization, centering, and differencing of MACH2.

17. Document Analysis a. Descriptors

Magnetohydrodynamics
Plasmas (Physics),
Simulation
Two-Dimensional,
MACH2 Computer Code
Hydrodynamic Codes

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group 20/09

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Cleared for public release; AFWL/PA 87-626 on 22 December 1987.	Unclassified	96
	20. Security Class (This Page) Unclassified	22. Price

CONTENTS

CONTENTS (Concluded)

FIGURES

FIGURES (Concluded)

1. INTRODUCTION

This report is a incomplete draft of a paper describing the physics, algorithms, and numerical methods used in the two-dimensional magnetohydrodynamic simulation code MACH2. It is being released in this form so that any aid it might provide to users of the code will be available as soon as possible. The author hopes that the final version of this report will be improved by this early review, and will gladly receive any comments or reports of errors which the reader might care to send to him.

The proposed outline for missing sections of the report is as follows:

Section

10. ITERATION PROCEDURES

- 10.1 JACOBI ITERATION
- 10.2 MULTIGRID ACCELERATION
- 10.3 THERMAL DIFFUSION
- 10.4 MAGNETIC DIFFUSION
- 10.5 CORRECTION OF NON-SOLENOIDAL DEVIATION
- 10.6 LAGRANGIAN HYDRODYNAMICS

11. IMPLEMENTATION

- 11.1 DATA STRUCTURE
  - 11.1.1 Geometry Description
  - 11.1.2 Spatially Dependent Physical Quantities
- 11.2 BOUNDARY CONDITIONS
  - 11.2.1 Boundary Condition Control
  - 11.2.2 Application by Stride
  - 11.2.3 Use of BCPOINT and BCPNTRS
- 11.3 DATA FLOW BY PHYSICAL PROCESS

In addition, material will be added to the sections dealing with time and space differencing to describe the recently added code which ensures that the poloidal magnetic field remains divergence free. In addition,

appendices connecting the theoretical descriptions of this report with the actual subroutines and variables of the code will be included in the final version of this report.

Even in this draft, the author wishes to acknowledge the dependence of the development of MACH2 on the contributions and strong efforts of many others. Without the technical foundation and help provided by Jerry Brackbill, this effort could not even have been attempted. He provided the source code for MOQUI, out of which MACH2 grew. The faith and support of Bob Reinovsky, Jim Degnan, and Bill Baker was as inspirational as it was essential. Moreover, considering how seldom experimentalists are willing to support applied theoretical efforts such as this, it was extraordinary. Their foresight and wisdom in this matter is a clear example for others to emulate. The importance of the technical advice and moral support given by Jim Buff and Norm Roderick can not be overstated. Their physical insights and experience with other simulation codes often provided the keys to unravel particularly knotty puzzles. Peter Turchi played a special role which must be mentioned. When the development of MACH2 entered that most difficult stage where the code ran, but its answers were not yet trusted, his insight and understanding of magnetohydrodynamics were an essential part of the process of persuasion. In the end, the success of the development effort hinged on the skills and efforts of those who actually wrote and debugged code. Bob Peterkin deserves most of the credit for the physics code while to Tony Giancola goes credit for the graphics, restart, and system interface code. Both made significant and substantial contributions in all phases of the development from the design stage through initial applications. The author is proud to be associated with this most capable team.

2. PHYSICAL MODEL

The physical models in the code today are largely those that are required to generate believable solutions to fast imploding liner or plasma flow switch problems. Specifically, the code includes ideal MHD, resistive diffusion of magnetic field and consequent joule heating, thermal diffusion, and radiation cooling. The differential equations expressing the conservation laws relevant to these processes will be described in detail below. In what follows d/dt represents the convective derivative.

2.1 THE MHD ELECTROMAGNETIC FIELD EQUATIONS

As explained in Jackson (Ref. 1), plasma phenomena with characteristic times much longer than the plasma oscillation period can be analyzed by neglecting the displacement current; this is the MHD approximation. In rationalized MKSA units then, the MHD fields are described by

$$\bar{\nabla} \times \bar{E} + \frac{\partial \bar{B}}{\partial t} = 0$$

$$\bar{\nabla} \times \bar{B} = \mu_o \bar{J} \qquad (1)$$

supplemented by an equation relating $\bar{J}$ and $\bar{E}$ that depends on the properties of the medium, i.e., Ohm's law

$$\bar{E} = \eta \bar{J} - \bar{v} \times \bar{B}$$

The electromagnetic units used in MACH2 are selected so that $\mu_o = 1$ and are related to the MKSA units by the following equations:

$$\bar{E}_{M2} = \sqrt{\frac{1}{\mu_{oMKSA}}}\ \bar{E}_{MKSA} \qquad \bar{B}_{M2} = \sqrt{\frac{1}{\mu_{oMKSA}}}\ \bar{B}_{MKSA}$$

$$\bar{J}_{M2} = \sqrt{\mu_{oMKSA}}\ \bar{J}_{MKSA} \qquad \eta_{M2} = \sqrt{\frac{1}{\mu_{oMKSA}}}\ \eta_{MKSA} \qquad (2)$$

In this system of units, the magnetic energy density is

$$P_m = B_{M2}^2 \quad /2 \tag{3}$$

and has units of Joules/meter^3. Henceforth, the subscript M2 will be suppressed, and all equations written in these units. Thus, the field equations in MACH2 are

$$\bar{\nabla} \times \bar{E} + \frac{\partial \bar{B}}{\partial t} = 0$$

$$\bar{\nabla} \times \bar{B} = \bar{J} \tag{4}$$

Application of Ohm's law for the moving fluid yields

$$\frac{\partial \bar{B}}{\partial t} = - \bar{\nabla} \times \left[(\eta \bar{\nabla} \times \bar{B}) - (\bar{v} \times \bar{B}) \right] \tag{5}$$

This can be rewritten as

$$\frac{d\bar{B}}{dt} = - \bar{\nabla} \times (\eta \bar{\nabla} \times \bar{B}) - (\bar{B}\, \bar{\nabla} \cdot \bar{v} - \bar{B} \cdot \nabla \bar{v}) \tag{6}$$

where the left-hand side represents the field transport, the first term of the right-hand side represents diffusion of field, and the second, the modification of field intensity by the divergence of the velocity field transverse to $\bar{B}$.

The resistivity used consists of two terms: one is classical and due to particle-particle interactions, the other is non-classical (anomalous) and intended to model particle interactions with the turbulent fields. The classical resistivity may be taken from a SESAME format table or computed from a Spitzer-like formula; the anomalous resistivity is always computed from a formula based on the ion acoustic frequency.

2.2 THE MHD MOMENTUM EQUATION

The compressible Navier-Stokes equations supplemented by the addition of the electromagnetic force are

$$\rho \frac{d\bar{v}}{dt} = -\bar{\nabla}p + \bar{J} \times \bar{B} + \bar{\nabla} \cdot \bar{\bar{\sigma}} \tag{7}$$

It is possible, and convenient, to write the entire right-hand side as the divergence of a tensor:

$$(-\bar{\nabla}p + \bar{J} \times \bar{B} + \bar{\nabla} \cdot \bar{\bar{\sigma}})_i = \frac{\partial}{\partial x_j} \left[(-p - \frac{1}{2}|B|^2)\delta_{ij} + B_i B_j + \sigma_{ij} \right]$$

$$= (\nabla \cdot \bar{\bar{S}})_i \tag{8}$$

2.3 THE CONTINUITY EQUATION

Conservation of mass is represented by

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \bar{v} \tag{9}$$

2.4 THE ENERGY EQUATION

If we let e be the internal energy per unit mass, then the energy equation in MACH2 is

$$\rho \frac{de}{dt} = -p\bar{\nabla} \cdot \bar{v} + (n\bar{\nabla} \times \bar{B}) \cdot \bar{\nabla} \times \bar{B}$$

$$+ (\bar{\bar{\sigma}} \cdot \bar{\nabla}) \cdot \bar{v} - \nabla \cdot \bar{F}_{diff} - \dot{q}_{rad} \tag{10}$$

and includes the flow work, Joule heating due to diffusion of magnetic field, diffusive transport of energy by thermal conduction or equilibrium

diffusion of radiation, and radiation cooling. The thermal conduction flux is limited to some fraction of the free-streaming flux which would be achieved if all the plasma on the high temperature side moved opposite to the gradient at the local average thermal velocity. The limit on the thermal flux is actually applied by limiting the thermal conductivity. Let k_{max} be the maximum allowable value of the thermal conductivity determined by

$$k_{max} = \frac{C_1 n_e T^{3/2}}{|\bar{\nabla} T|} \tag{11}$$

Then, if the thermal conductivity determined by local plasma conditions is k_p, the effective, or limited, thermal conductivity k_e is determined by

$$k_e = \frac{k_p \, k_{max}}{k_p + k_{max}} \tag{12}$$

The thermal conductivity k_r which gives an energy flux equivalent to that due to diffusion of a radiation field in equilibrium with the plasma is

$$k_r = \frac{C_2 T^3}{\kappa \rho} \tag{13}$$

The diffusive flux of energy is then

$$F_{diff} = -(k_e + k_r) \, \bar{\nabla} T \tag{14}$$

The radiation cooling rate is taken to be

$$\dot{q}_{rad} = -C_3 \, \kappa T^4 \tag{15}$$

This is applicable when the plasma is thin, and the radiation spectrum may be assumed to be Planckian.

2.5 THE EQUATIONS OF STATE AND TRANSPORT PROPERTIES

All of the equations of state quantities, and some of the transport properties, are determined using EOSPAC (Ref. 2) to do table look-up in the Los Alamos SESAME Equation of State Library (Ref. 3). The models used to generate these tables are among the best known; however, there are regions in density and temperature space where interpolation between models has been used. In general, those regions are there because no tractable numerical model exists which applies; hence, no simple analytic model is likely to be superior!

3. BOUNDARY CONDITIONS

Modeling complex experiments requires a wide variety of boundary treatments even for this modest list of physics. The mathematical expression of these will be detailed below, organized by the physics to which they apply. Details of a numerical or code implementation nature will be given in later sections.

The philosophy of all boundary condition application in the code is that the conditions described are the limit of conditions in the fluid as the boundary is approached. Thus, the boundary of the computational region is the edge of the region in which physical modeling is done. Quantities outside this region are assumed known or related to quantities inside by, at most, simple instantaneous geometric statements. Specifically, no time integration of spatially varying quantities is carried on outside the problem boundaries.

Figure 1 shows a small section of problem boundary and gives the geometric quantities which will be used in the description of boundary conditions to follow.

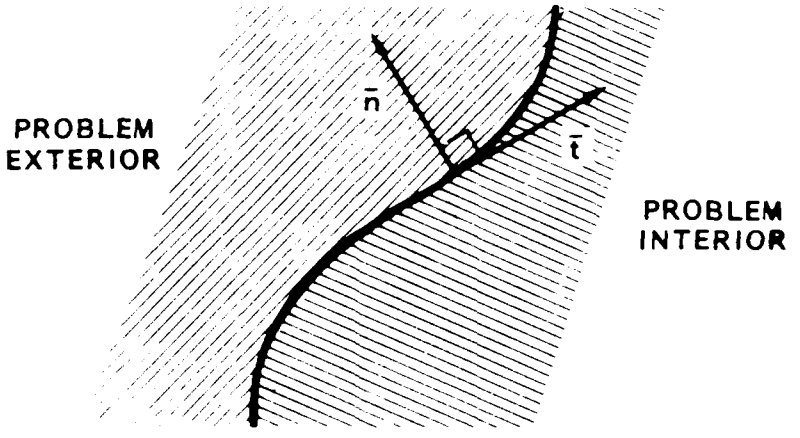


Figure 1. Boundary geometric quantities.

The tangent vector field, $\bar{t}$, and the normal vector field, $\bar{n}$, will be taken to be normalized so that

$$|\bar{n}| = |\bar{t}| = 1 \tag{16}$$

No convention will be adopted for the sense of $\bar{n}$ and $\bar{t}$; it will be specified only if circumstances require it.

The normal and tangent vectors always are taken as lying in the computational plane. That is

$$\bar{n} \cdot \hat{e}_\theta = 0$$

$$\bar{t} \cdot \hat{e}_\theta = 0 \tag{17}$$

for cylindrical symmetry and

$$\bar{n} \cdot \hat{e}_z = 0$$

$$\bar{t} \cdot \hat{e}_z = 0 \tag{18}$$

for planar symmetry.

Discussion in the remainder of this section will be restricted to the cylindrical case. In all cases, the results of that discussion may be extended to planar symmetry by the transformation of

$$\begin{aligned} r &\rightarrow x \\ z &\rightarrow y \\ \theta &\rightarrow z \end{aligned} \tag{19}$$

for and the substitution r = 1 where r is used algebraically and not as a coordinate.

3.1 MAGNETIC FIELD BOUNDARY CONDITIONS

The boundary conditions used in the code represent those appropriate within the fluid at the surface of idealized conductors and insulators, at a symmetry boundary, or at the axis of cylindrical symmetry.

3.1.1 Perfect Conductor--As shown in Jackson (Ref. 4), the surface charge and current instantly nullify any non-zero field in a perfect conductor. Since the normal component of $\overline{B}$ and the tangential component of $\overline{E}$ must be continuous through the surface, it follows that as the surface is approached from outside the conductor the following are true in the limit:

$$\overline{n} \cdot \overline{B} = 0$$

$$\overline{n} \times \overline{E} = 0 \tag{20}$$

Since the magnetic field is the fundamental quantity in the code the second equation above must be transformed into an equation involving $\overline{B}$. The appropriate equation is

$$\overline{n} \times \overline{E} = \overline{n} \times (\eta \overline{J} - \overline{v} \times \overline{B}) = 0 \tag{21}$$

The resistivity will be considered scalar here. Thus, $\overline{n} \times (\eta \overline{J}) = \eta (\overline{n} \times \overline{J})$. Since $\overline{n} \times (\overline{v} \times \overline{B}) = (\overline{n} \cdot \overline{B})\overline{v} - (\overline{n} \cdot \overline{v})\,\overline{B}$ and $\overline{n} \cdot \overline{B} = 0$ we have

$$\eta\, (\overline{n} \times \overline{J}) + (\overline{n} \cdot \overline{v})\, \overline{B} = 0 \tag{22}$$

If the boundary is such that $\overline{n} \cdot \overline{v} = 0$, i.e., it is either a free slip or a no slip boundary, then this becomes

$$\bar{n} \times \bar{J} = 0$$

or

$$J_\theta = 0$$

and

$$n_r J_z - n_z J_r = 0 \tag{23}$$

Written in terms of $\bar{B}$, the coordinatized version of this in cylindrical coordinates (where $\partial/\partial\theta = 0$) is

$$\left.\begin{array}{l} \bar{n} \cdot \bar{B} = 0 \\ \quad \text{and} \\ (\bar{\nabla} \times \bar{B})_\theta = 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} \bar{n} \cdot \bar{B} = 0 \\ \quad \text{and} \\ \bar{n} \cdot \bar{\nabla} (\bar{t} \cdot \bar{B}) = 0 \end{array}\right. \tag{24}$$

and

$$\frac{1}{r} \bar{n} \cdot \bar{\nabla} (rB_\theta) = 0 \tag{25}$$

It is interesting to contemplate what effect a porous perfect conductor (one for which $n \cdot v \neq 0$) would have on the field adjacent to it, but in the range of energy densities of interest to us, it would probably plug up immediately with ablated mass. Thus, perfect conductors are modeled by Equations 23 through 25.

3.1.2 Perfect Insulator--No current flows into or out of a perfectly insulating wall. Thus, $\bar{n} \cdot \bar{J} = 0$ at the wall. Since

$$\begin{aligned} \bar{n} \cdot \bar{J} &= (n_r, 0, n_z) \cdot \left(- \frac{\partial B_\theta}{\partial z}, \frac{\partial B_r}{\partial z} - \frac{\partial B_z}{\partial r}, \frac{1}{r} \frac{\partial}{\partial r} (rB_\theta) \right) \\ &= \frac{1}{r} (n_z, 0, -n_r) \cdot \left(\frac{\partial}{\partial z}(rB_\theta), 0, \frac{\partial}{\partial r} (rB_\theta) \right) \\ &= \frac{1}{r} t \cdot \nabla (rB_\theta) \end{aligned} \tag{26}$$

where $\bar{t} \cdot \hat{e}_\theta = 0$ and $\bar{t}$ is tangent to the wall, it is clear that rB_θ is constant along an insulating wall. The constant, of course, is proportional to the total current flowing through the (possibly empty) center of the (possibly) annular surface of revolution forming the wall. It can, therefore, vary from one insulating boundary to another.

While $\bar{n} \cdot \bar{J} = 0$ is quite restrictive on B_θ, it has no content for B_r and B_z in azimuthally symmetric geometry, since B_r and B_z affect only J_θ and $n_\perp \hat{e}_\theta$ due to the cylindrical symmetry. Thus, the appropriate boundary condition for B_r and B_z at a perfect insulator is that they be specified by external considerations. For example, they may be determined from specified current flowing in conductors such as Helmholtz coils outside the region of interest. Figure 2 shows such a configuration.

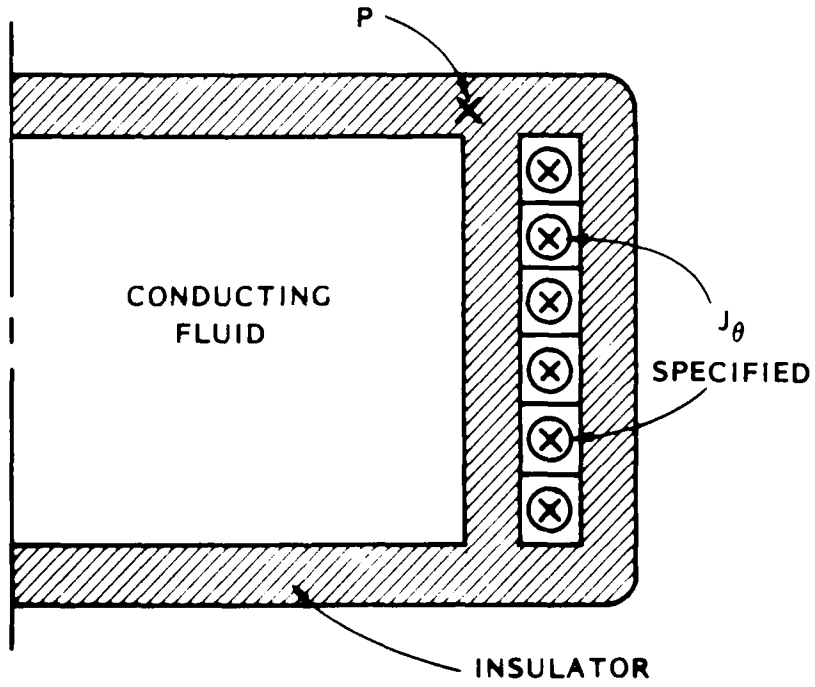


Figure 2. A problem with insulating boundary conditions.

The magnetic field at boundary points such as P may be determined by applying the Biot-Savart law to the segmented conducting loops. This field will diffuse into the conducting fluid, and induce current in it.

3.1.3 Symmetry Surface--It is often possible to take advantage of symmetry in the problem geometry to eliminate half the problem. Usually, this is done by applying a boundary condition which implies that the conditions just outside the problem region are a mirror image of those inside it. For the magnetic field, the appropriate conditions may be derived from the requirements that tangential components of (J_r, J_z) and (B_r, B_z) be zero on such a boundary, and the normal components of (B_r, B_z) be continuous. The first of these requires that

$$\frac{1}{r}\ \bar{n} \cdot \nabla\ (rB_\theta) = 0 \tag{27}$$

while the other two require

$$\bar{t} \cdot \bar{B} = 0$$

$$\bar{n} \cdot \bar{\nabla}\ (\bar{n} \cdot \bar{B}) = 0 \tag{28}$$

where we make the assumption then $\bar{n}$ is extended off the boundary so then $\bar{n} \cdot \overline{\nabla n} = 0$.

3.1.4 The Axis of Cylindrical Symmetry--At $r = 0$, the appropriate conditions are that

$$B_\theta = 0$$

$$B_r = 0$$

$$J_\theta = 0 \tag{29}$$

Since

$$J_\theta = \frac{\partial B_r}{\partial z} - \frac{\partial B_z}{\partial r} \tag{30}$$

Equations 29 are completely equivalent to

$$B_\vartheta = 0$$

$$B_r = 0$$

$$\frac{\partial B_z}{\partial r} = 0 \tag{31}$$

3.2 HYDRODYNAMIC BOUNDARY CONDITIONS

The boundary conditions applied to the hydrodynamic quantities represent those conditions within the fluid at the surface of walls, at inlets, and outlets where external conditions are presumed known, at inlets and outlets where external conditions are the same as the internal conditions adjacent to the surface, and at the axis of cylindrical symmetry.

3.2.1 Walls--At a boundary representing an impermeable wall, the relevant boundary condition is either free slip

$$\bar{n} \cdot \bar{v} = 0 \tag{32}$$

or no slip

$$\bar{v} = 0 \tag{33}$$

3.2.2 Inlets and Outlets--At an inlet the properties which determine the stress tensor and the convective fluxes must be specified. There the boundary conditions must include

$$e = e_{inlet} \quad \text{or} \quad T = T_{inlet}$$

and

$$\rho = \rho_{inlet} \tag{34}$$

They must also include

$$\bar{v} = \bar{v}_{inlet} \tag{35}$$

The same quantities must be restricted at an outlet:

$$\bar{n} \cdot \bar{\nabla} e = 0$$

$$\bar{n} \cdot \bar{\nabla} \rho = 0$$

$$\bar{n} \cdot \bar{\nabla} \bar{v} = 0 \tag{36}$$

3.2.3 The Axis of Cylindrical Symmetry--At $r = 0$, the appropriate conditions are

$$v_r = 0$$

$$v_\theta = 0 \tag{37}$$

3.3 THERMAL CONDUCTION BOUNDARY CONDITIONS

The conditions applied to the temperature at boundaries are of two types: conduction to a fixed temperature reservoir of infinite heat capacity, and no flux at all. In the former case, the condition is

$$T = T_{wall} \tag{38}$$

and in the latter it is

$$\bar{n} \cdot \bar{\nabla} T = 0 \tag{39}$$

4. GEOMETRIC PROBLEM CLASS

The geometries of real experimental devices are usually much more complex than the idealized geometries that motivate the experimental design. Yet these deviations from the ideal geometry may have significant quantitative and qualitative effects on the results. It is thus essential that a two-dimensional code be capable of solving problems in domains which closely model the real device. Further, some interesting experimental effects cannot be seen at all in idealized geometry. Exploration of them with simulation codes limited to domains with idealized geometry is clearly impossible.

MACH2 was designed to handle any of a broad geometric class of domains without code modification. In this section, that geometric class will be described, and some complex experimental domains shown to be admissible in that class.

A subset B of $\mathbf{R}^2$ given by

$$B_{\underline{a},\underline{b}} = \{(x,y)\in \mathbf{R}^2 \mid a_x \leq x \leq b_x,\ a_y \leq y \leq b_y\} \tag{40}$$

where $\underline{a} = (a_x, a_y)$ and $\underline{b} = (b_x, b_y)$, is a rectangle and will be referred to as a block. Recall that a diffeomorphism is a smoothly invertible smooth mapping, that is, a sufficiently differentiable function

$$f : U \subset \mathbf{R}^2 \rightarrow \mathbf{R}^2 \tag{41}$$

for which there exists sufficiently differentiable function

$$g : W \subset \mathbf{R}^2 \rightarrow \mathbf{R}^2 \tag{42}$$

called the inverse of f, which satisfies

$$g(f(w)) = u \qquad \text{for all } u\in U \tag{43}$$

and

$$f(g(w)) = w \qquad \text{for all } w \in W \tag{44}$$

A region $B \subset \mathbf{R}^2$ which is the diffeomorphic image of a block will be called block-like.

The regions shown in Figure 3 are examples of block-like regions, which may be thought of as rectangles formed from rubber sheets which are deformed by the diffeomorphism. The examples in Figure 4 are also block-like except that the diffeomorphism is not one-to-one on the entire block, but is multiple valued on at least one edge. Thus, the inverse mapping is singular at the sector center and at the joining cut in the annulus. Such regions will be called singular block-like regions.

The geometric objects which MACH2 is designed to treat as domains consist of a particular kind of union of block-like regions. They are best described as (almost) diffeomorphic images of a union of blocks

$$\bigcup_{i=1}^{n} B_{\underset{\sim}{a}_i, \underset{\sim}{b}_i} \tag{45}$$

with the $\underset{\sim}{a}_i$ and $\underset{\sim}{b}_i$, chosen so that any two blocks which meet, do so only along an entire edge or at a vertex. Such a collection of blocks will be called a block complex. Figure 5 shows some examples of block complexes. They resemble subsets of checkerboard consisting only of whole squares. It is natural to call the diffeomorphic image of a block complex, a block-like complex.

This class is quite broad. It is more than broad enough to include realistic experimental configurations. Figure 6 shows two configurations in interest which are within the geometric class described above. In the present version MACH2, corners of a logical block must meet other logical blocks only at their corners. Further, any change in boundary condition,

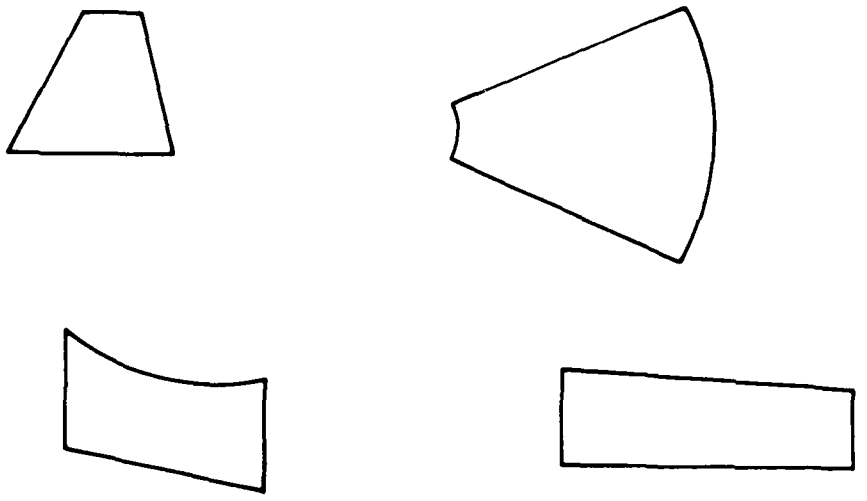


Figure 3. Block-like regions.

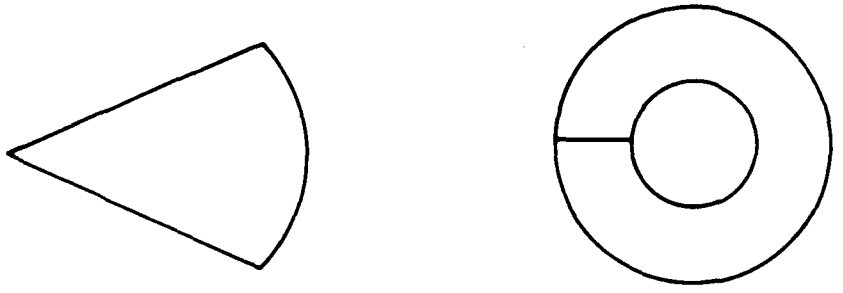
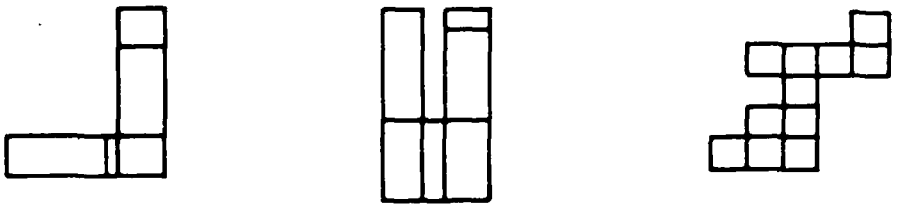
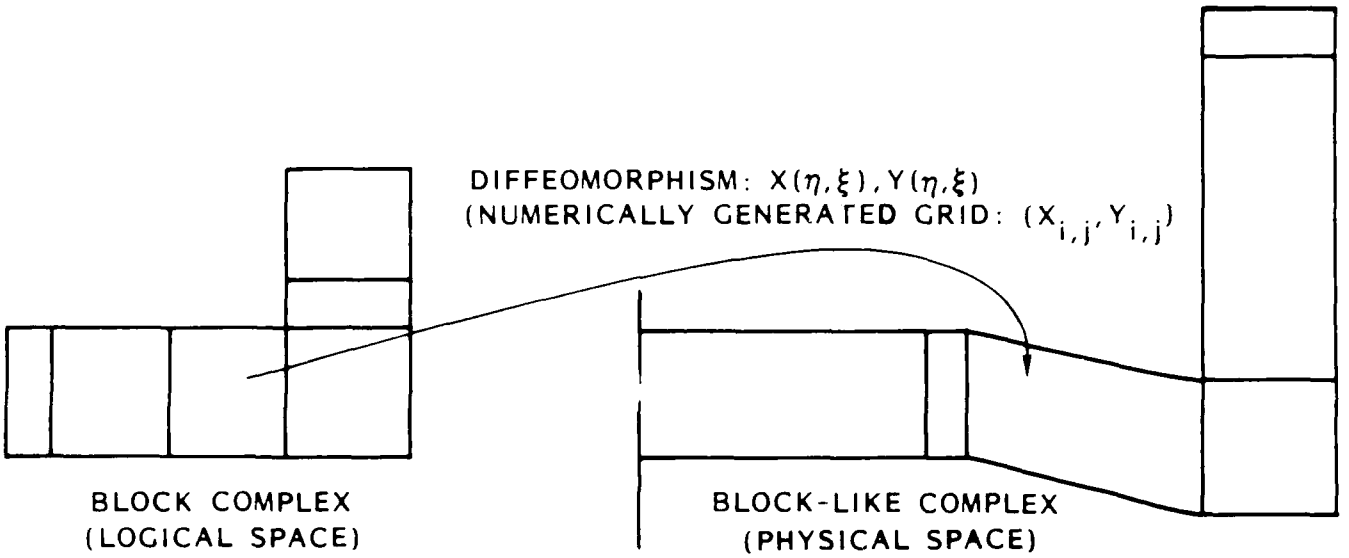


Figure 4. Singular block-like regions.

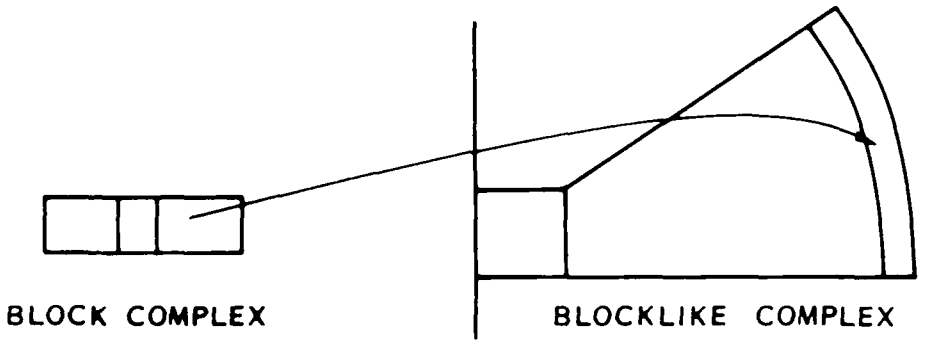


AMRC-R-874

Figure 5. Block complexes.



a) PLASMA FLOW SWITCH



b) SPHERICAL LINER IMPLOSION

AMRC-R-874

Figure 6. Real experimental configurations as block-like complexes.

whether physical or grid, any sharp change of direction of the region boundary, or any change in uniform initial condition, must occur at a block corner or edge. Thus, some of the block interfaces shown in Figure 5 are required for non-geometric reasons.

5. COORDINATE SYSTEM

In MACH2 the physical quantities are computed on a coordinate system which moves arbitrarily in space. Since such coordinate systems can be made to remain fixed in space or to move with the fluid, they are called Arbitrary Lagrangian-Eulerian systems. Clearly, this name is unfortunate, and we will prefer "arbitrary coordinate system" or "arbitrary fluid description".

The principal advantages of such a description, described in References 5 and 6, are that some of the catastrophic coordinate distortion common with a Lagrangian description and the diffusive inaccuracy of the Eulerian description can be avoided. The principal disadvantage is that a prescription for the coordinate system motion must be supplied.

In MACH2, the coordinate system for a block-like complex is described by a collection of function pairs

$$\{(x_l(\eta,\xi,t)\ ,\ y_l(\eta,\xi,t)) \ \Big|\ (\eta,\xi) \in B_l,\ t \in \mathbf{R}^+,\ l = 1,\ldots,L\} \quad (46)$$

defined on the collection of L individual blocks, B_l, making up the block complex which is the domain. The coordinates (η,ξ) are just the continuous extensions of the integer array coordinates (i,j) that will make their appearance upon discretization. All of the physical quantities are also described as functions of $(\eta,\xi) \in B_l$, so that the full problem is actually given by a parametric description. That is $\overline{B}$, ρ, and $\overline{v}$ are stored as functions of l, and of (η,ξ) rather than (x,y).

In the next two subsections, the range of choices of coordinate systems available in MACH2 and the effects of this choice on the physical model, as represented by the partial differential equations of Section 2, will be described. Coordinates in the logical domain (l,η,ξ) will be represented by $\overline{\xi}$.

5.1 COORDINATE SYSTEM MOTION

The coordinate system $\bar{x}(\bar{\xi},t)$ is allowed to move according to

$$\frac{d\bar{x}}{dt}(\bar{\xi},t) = \alpha\left(\bar{x}_I(\bar{\xi},t) - \bar{x}(\bar{\xi},t)\right) + \beta\bar{v} \tag{47}$$

where $\bar{v}$ is the fluid velocity, $\bar{x}_I$ is an "ideal" coordinate created by a numerical grid generation procedure which we will describe shortly, and the control functions $\alpha(\bar{\xi})$ and $\beta(\bar{\xi})$ are user specified.

The Langrangian coordinate system results if $\alpha \equiv 0$ and $\beta \equiv 1$, while the Eulerian coordinate system appears if both α and β are identically zero. Sensible values for α lie in the range $0 \leq \alpha \leq 1/T$, where T is some relaxation time, while β must be either 0 or 1. Thus, α is a relaxation parameter telling how quickly the actual coordinate system moves toward the ideal coordinate system.

5.2 THE IDEAL COORDINATE SYSTEM

The ideal coordinate system is generated by solving a Brackbill-Saltzman variational problem on the block complex domain subject to user specified external and internal boundary conditions and using a weight function selected from a standard family by the values of user specified parameters. The details of the variational problem and its Euler equations are described in Brackbill (Ref. 7), and their implementation for block-like complexes is described in Reference 8. For completeness, a brief description will be given here.

Let

$$d_{\bar{\xi}}\bar{x} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix} \tag{48}$$

so that J, the Jacobian of the mapping $(\eta,\zeta) \longrightarrow (x,y)$, is given by

$$J = \det \left(d_{\bar{\xi}} \bar{x} \right) \tag{49}$$

Then, the variational problem for the mesh is stated as

$$\delta \int_D \mathscr{S}\left(d_{\bar{\xi}} \bar{x}_I \right) + \lambda_v \mathscr{V}\left(d_{\bar{\xi}} \bar{x}_I, \bar{x} \right) + \lambda_o \mathscr{O}\left(d_{\bar{\xi}} \bar{x}_I \right) \; dA = 0 \tag{50}$$

where λ_v and λ_o are user specified parameters, D is the block complex domain, and the nonlinear operators $\mathscr{S}$, $\mathscr{V}$, and $\mathscr{O}$ are local measures of the mesh smoothness, its nearness to desired concentration, and its orthogonality, respectively. These operators are given by

$$\mathscr{S}\left(d_{\bar{\xi}} \bar{x} \right) = \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] / J$$

$$\mathscr{V}\left(d_{\bar{\xi}} \bar{x}, \bar{x} \right) = w(x,y) \, J^2$$

$$\mathscr{O}\left(d_{\bar{\xi}} \bar{x} \right) = \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right)^2 \tag{51}$$

where w is a user specified volume control function. Thus, for $\lambda_o = \lambda_v = 0$ the solution to this variational problem is, in some sense, the smoothest coordinate system filling the physical region; whereas, for $\lambda_o = 1$ and $\lambda_v = 0$ it is some blend of a smooth coordinate system and an orthogonal one. Of course, for $\lambda_v \neq 0$ the character of w will have an effect on the solution to the variational problem, so if w depends on the current state of the simulation variables, then the coordinate system may dynamically adapt as the simulation proceeds.

The role of the function w in controlling the adaptivity of the mesh can be made more precise. The Euler equations for Equation 49 with $\lambda_v = \infty$ are

$$\frac{\partial w}{\partial x} J^2 + 2 \left[(wJ)_\xi \, y_\eta - (wJ)_\eta y_\xi \right] = 0$$

$$\frac{\partial w}{\partial y} J^2 + 2 \left[x_\xi (wJ)_\eta - x_\eta (wJ)_\xi \right] = 0 \qquad (52)$$

If we suppose that $(x(y,\xi), y(\eta,\xi))$ satisfying these has been found, let $\mathcal{J}(x,y) = x_\xi(\eta,\xi)\, y_\eta(\eta,\xi) - x_\eta(\eta,\xi)\, y_\xi(\eta,\xi)$, and transform Equation 52 using

$$\frac{\partial}{\partial \xi} = x_\xi \frac{\partial}{\partial x} + y_\xi \frac{\partial}{\partial y}$$

$$\frac{\partial}{\partial \eta} = x_\eta \frac{\partial}{\partial x} + y_\eta \frac{\partial}{\partial \eta} \qquad (53)$$

then we obtain

$$\frac{\partial}{\partial x} \left(w \mathcal{J}^2 \right) = \frac{\partial}{\partial y} \left(w \mathcal{J}^2 \right) = 0 \qquad (54)$$

Thus, the Jacobian of the coordinate system defined by Equation 52 must be proportional to $w^{-1/2}$. Since the Jacobian of the coordinate system is approximately the cell area for the mesh, cells will be smaller where w is large, so long as $\lambda_v > 0$.

For λ_v of order unity, the terms in the Euler Equations for Equation 50 which are derived from $\mathcal{S}$ act to smooth the coordinate system and reduce the influence of the weight function. The resulting coordinate system is thus a blend of those determined by the operators in Equation 51.

5.3 THE MATERIAL DERIVATIVE

The only terms of the physical model of Section 2 affected by the choice of coordinate system are those involving the material derivative. The material derivative for a quantity in Lagrangian description is

$$\frac{df}{dt} = \frac{\partial f_l}{\partial t} \qquad (55)$$

while for an Eulerian description it is

$$\frac{df}{dt} = \frac{\partial f_e}{dt} + \bar{v} \cdot \bar{\nabla} f_1 \tag{56}$$

For the arbitrary fluid description, it is

$$\frac{df}{dt} = \frac{\partial f_a}{\partial t} + (\bar{v}-\bar{v}_c) \cdot \bar{\nabla} f_a \tag{57}$$

where $\bar{v}_c$ is the velocity of the coordinate system defined as the velocity of points in physical space represented with fixed coordinates. Clearly, the special cases $\bar{v}_c = \bar{v}$ and $\bar{v}_c = 0$ produce Equation 55 and Equation 56 from Equation 57. Equation 57 is derived from Equation 55 by noting that

$$f_a(\eta,\xi,t) = f_e[x(\eta,\xi,t), y(\eta,\xi,t), t] \tag{58}$$

and differentiating with respect to t to obtain

$$\frac{\partial f_a}{\partial t} = \frac{\partial f_e}{\partial t} + x_t \frac{\partial f_e}{\partial x} + y_t \frac{\partial f_e}{\partial y} \tag{59}$$

Since $(x_t, y_t) = \bar{v}_c$, solving Equation 59 for $\partial f_e/\partial t$ and substituting the result into Equation 55 produces Equation 57.

The quantities of which material derivatives are required in the model include ρ, e, $\bar{B}$, and $\bar{v}$. It is crucial to the performance of the code that these be performed in a way which conserves certain integral quantities. The details of the spatial differencing by which that is accomplished will be covered in Section 7. The equivalent integral statements of the differential equation

$$\frac{d\phi}{dt} = 0 \tag{60}$$

differenced there will be derived here.

If $R(t)$ is a moving region of space, then

$$\frac{\partial \phi}{\partial t} + \bar{\nabla} \cdot (\phi \bar{v}) = 0 \tag{61}$$

implies

$$\frac{\partial}{\partial t} \int_{R(t)} \phi dV = \int_{\partial R(t)} \phi (\bar{v}_s - \bar{v}) \cdot \bar{n} \, dA \tag{62}$$

where ϕ is described in Eulerian coordinates, $\bar{v}$ is the fluid velocity, and v_s is the velocity of the surface. Conversely, if Equation 62 is true for all regions $R(t)$ then Equation 61 is a consequence.

To illustrate the equivalence of these two statements, we must parameterize the moving region $R(t)$ by $\bar{x}(\bar{s},t)$

$$\bar{x} : \Omega \longrightarrow R(t) \tag{63}$$

where $\bar{x},(\bar{s},0) = s$, so that $\Omega = R(0)$. Then

$$\int_{R(t)} \rho dV = \int_{\Omega} \rho(y,t) \det (d_{\bar{s}} \bar{y}) \, d\bar{s} \tag{64}$$

in which $d_{\bar{s}} \bar{y}$ is the Jacobian matrix

$$(d_{\bar{s}} \bar{y})_{ij} = \frac{\partial y_i}{\partial s_j} \tag{65}$$

so that $\det (d_{\bar{s}} \bar{y})$ is the Jacobian. From Equation 64

$$\frac{\partial}{\partial t} \int_{R(t)} \phi dV = \int_{\Omega} \frac{\partial \phi}{\partial t} + \frac{\partial \bar{y}}{\partial t} \cdot \nabla \phi + \phi \frac{\partial}{\partial t} (\det d_{\bar{s}} \bar{y}) \, d\bar{s} \tag{66}$$

Now,

$$\frac{\partial}{\partial t}\left(\det d_{\bar{s}}\bar{y}\right) = \left(\nabla \cdot \bar{v}_s\right)\left(\det d_{\bar{s}}\bar{y}\right) \tag{67}$$

so

$$\frac{\partial}{\partial t}\int_{R(t)} \phi dV = \int_{R(t)} \frac{\partial \phi}{\partial t} + \bar{\nabla}(\phi \bar{v}_s)\, dV \tag{68}$$

Substituting $\bar{v} + (\bar{v}_s - \bar{v})$ for $\bar{v}_s$ and applying $\partial\phi/\partial t + \bar{\nabla}(\phi\bar{v}) = 0$ yields

$$\frac{\partial}{\partial t}\int_{R(t)} \phi dV = \int_{R(t)} \bar{\nabla} \cdot ((\bar{v}_s - \bar{v})\phi)\, dV \tag{69}$$

An application of Stokes Theorem then produces Equation 62. Proof of the converse uses the same algebra and the fact that

$$\int_R f\, dV = 0 \qquad \text{for all } R \tag{70}$$

implies that f must be zero if it is continuous.

By applying Equation 61 repeatedly with $\phi = \rho$, ρe, $\rho\bar{v}$, and $\rho\bar{B}$, it can be shown that the system

$$\frac{d\rho}{dt} = -\rho\, \nabla \cdot v$$

$$\frac{de}{dt} = 0 \qquad \frac{d\bar{v}}{dt} = \frac{d\bar{B}}{dt} = 0 \tag{71}$$

is equivalent to

$$\frac{\partial}{\partial t}\int_{R(t)} \rho dV = \int_{\partial R(t)} \rho\ (\bar{v}_s - \bar{v}) \cdot \bar{n}\ dA$$

$$\frac{\partial}{\partial t}\int_{R(t)} \rho e dV = \int_{\partial R(t)} \rho e\ (\bar{v}_s - \bar{v}) \cdot \bar{n}\ dA$$

$$\frac{\partial}{\partial t}\int_{R(t)} \rho \bar{v} dV = \int_{\partial R(t)} \rho \bar{v}\ (\bar{v}_s - \bar{v}) \cdot \bar{n}\ dA$$

$$\frac{\partial}{\partial t}\int_{R(t)} \rho \bar{B} dV = \int_{\partial R(t)} \rho \bar{B}\ (\bar{v}_s - \bar{v}) \cdot \bar{n}\ dA \qquad (72)$$

It is clear that the first of Equations 72 is equivalent to the first of Equations 71. The second equation follows from expanding

$$\frac{\partial}{\partial t}(\rho e) + \bar{\nabla} \cdot (\rho e \bar{v}) = \rho \frac{de}{dt} + e\left(\frac{d\rho}{dt} + \rho \bar{\nabla} \cdot \bar{v}\right) \qquad (73)$$

The second of Equations 72 implies that the left-hand side of this is zero; the factor in parenthesis on the right-hand side is zero by the previous remark. Since ρ cannot be zero anywhere, the second of Equations 71 follows. The remainder follow in a similar manner.

6. TIME DIFFERENCING

The physical model described in the previous sections is solved numerically in arbitrary coordinates by a time-split, time-marching algorithm. The physical processes which are time-split are:

Radiation Cooling
Thermal and Equilibrium Radiation Diffusion
Resistive Diffusion
Lagrangian Hydrodynamics
Coordinate System Motion and Convective Transport

The resistive diffusion and the Lagrangian hydrodynamics are done with implicit time differencing, but the remainder are done with explicit differencing. The time-step is selected to maintain the stability of the calculation.

6.1 TIME-SPLITTING

Often known as fractional time-stepping, time-splitting is an approach to time advancing evolution equations. It consists of the sequential application of separate portions of the system of equations, rather than simultaneous application of the entire set. Thus, the conditions or state which result from the application of the first subset of equations or terms are used in the second, to compute the changes in state dictated by that piece or pieces of the governing physical model. In MACH2, for example, the magnetic field which results from the action of the resistive diffusion for the current time-step is used for the initial state for the action of the Lagrangian hydrodynamics for the same time-step. The inaccuracy of this time differencing can, of course, be made small by making the time-step sufficiently small. Later subsections of this section will describe precisely how the various equations, and in fact, even terms of equations are time-split.

6.2 EXPLICIT VERSUS IMPLICIT TIME DIFFERENCING

When an evolution equation of the form

$$\frac{\partial \chi}{\partial t} = F(\chi) \tag{74}$$

is solved numerically, the values of the dependent variables will only be computed at discrete time t_1, t_2,...t_k,.... For a finite difference code, the dependent variable χ is a vector with many components, one for each physical variable of interest (B_r, B_θ, B_z, ρ, u, v, w , x, y, e, and others as well in the present case) at each spatial location of interest i.e., at each grid point. If we use χ_n to stand for $\chi(t_n)$ one way to difference Equation 74 is

$$(\chi_{n+1} - \chi_n)/dt = F(\chi_n) \tag{75}$$

This is known as explicit time *differencing*, because the new time value χ_{n+1} is given explicitly as a function of the old time value χ_n by

$$\chi_{n+1} = \chi_n + dt\ F(\chi_n) \tag{76}$$

The iterative procedure described by the spatially differenced form of this equation can be unstable. That is, it may describe non-physical divergence from physically reasonable behavior. This certainly will happen if the linear part of Equation 76 amplifies disturbances with characteristic lengths of variation as small as the spatial descretization dx. This behavior is associated with the existence of eigenvalues of the linearization of Equation 76 which have magnitude greater than 1. Often the requirement that these eigenvalues be less than 1 in magnitude can be maintained by keeping dt small enough. The existence of such a stability limit on the time step is typical of explicit time differencing; implicit time differencing can be free of such limits.

Fully implicit time differencing for Equation 54 is given by

$$(x_{n+1} - x_n)/dt = F(x_{n+1}) \tag{77}$$

and is so-called because x_{n+1} is determined implicitly from x_n by the relation

$$x_{n+1} - dt\ F(x_{n+1}) = x_n \tag{78}$$

This must be solved for x_{n+1} to determine the new time values. In any interesting case, each equation represented by Equation 78 depends on many of the components of x_{n+1}, of which there may be hundreds. Numerical solution of such systems is typically very similar to the solution of the steady-state problem (derived from Equation 77 by taking the limit $dt \to \infty$)

$$F(x_\infty) = 0 \tag{79}$$

If the partial differential equations represented by Equation 74 are parabolic, such as the equation of thermal diffusion, then the steady-state problem will be elliptic, as will the time advance problem, Equation 77. Techniques for solving elliptic problems numerically such as successive over-relaxation, alternating direction implicit methods, multigrid methods, and preconditioned conjugate gradient methods can all be used. Since the operator F is often nonlinear, these methods all must be modified to include same iterative methods for the solution of nonlinear equations, usually similar to Newton's method.

The relationship of implicit or explicit time differencing to permissible time-step is determined by stability considerations in a very general framework. In either case, we must consider the linear part of the iteration operator L such that

$$\delta x_{n+1} = L\ \delta x_n \tag{80}$$

For explicit iteration we have $L = L_e \equiv I + dt\ d_x F$ from Equation 76; the implicit iteration, Equation 78, implies $L = L_e \equiv (I - dt\ d_x F)^{-1}$. Thus, if we write $\Lambda(M)$ for the spectrum of an operator M, we have the spectral mappings

$$\Lambda(L_e) = 1 + dt\ \Lambda(d_x F) \tag{81}$$

and

$$\Lambda(L_i) = \frac{1}{1 - dt\ \Lambda(d_x F)} \tag{82}$$

For parabolic Equations 74, $\Lambda(d_x F) \subset LHP = \{z \in C \mid R_e(z) < 0\}$; further if $d_x F$ is taken as referring to the discretized F, the $\Lambda(d_x F)$ is bounded since it has only finitely many members. Then Equation 66a implies that if dt is small enough, $\Lambda(L_e) < S(0,1)$. Here $S(z_o, r) = \{z \in C \mid |z-z_o| < r\}$. This is precisely the condition required for stability of the iteration, since no eigenvectors exist which are amplified by the iteration. However, it is clear that if dt is too large $\Lambda(L_e)$ will extend beyond S(0,1) and same disturbances will be amplified. Thus, explicit time-stepping requires a time-step control.

The case of implicit iteration is more interesting: Equation 82 shows that $\Lambda(d_x F) \subset LHP \Rightarrow \Lambda(L_i) \subset S(0,1)$ regardless of the spectral radius of $d_x F$ or of dt. This is because 1/(1-z), the spectral function, maps LHP into $S(1/2,1/2) \subset S(0,1)$. Hence, the iteration is stable at all time-steps, and no time-step control is required by stability considerations.

That this argument should be possible without assuming $d_x F$ to have constant coefficients is truly remarkable. It is probably possible to extend it to non-parabolic operators F as well.

In addition to describing the time-splitting, the remainder of Section 6 will describe the time dependence of the time advance equations used for each physical process modeled by the code to point out which are explicit

and which are not. Details of the iterative numerical solutions of those that are implicit will be reserved to Section 7.

For purposes of reference, we will think of the various stages intermediate to the major time-split processes as being at different times. The times will be referred to as t_I, t_R, t_T, t_D, t_H, and t_F. The time at the initiation of each cycle is t_I; after the radiation cooling, t_R; after thermal diffusion, t_T; after magnetic diffusion, t_D; after the Lagrangian hydrodynamics, t_H; and after the convective flux which finishes the cycle, t_F, which is the same as $t_I + dt$.

Figure 7 shows schematically which major physical processes of the main computational cycle advance each of the fundamental variables between those times. From it, one can see that density is only changed by the convective transport, while the internal energy is changed by all of the major processes in the cycle.

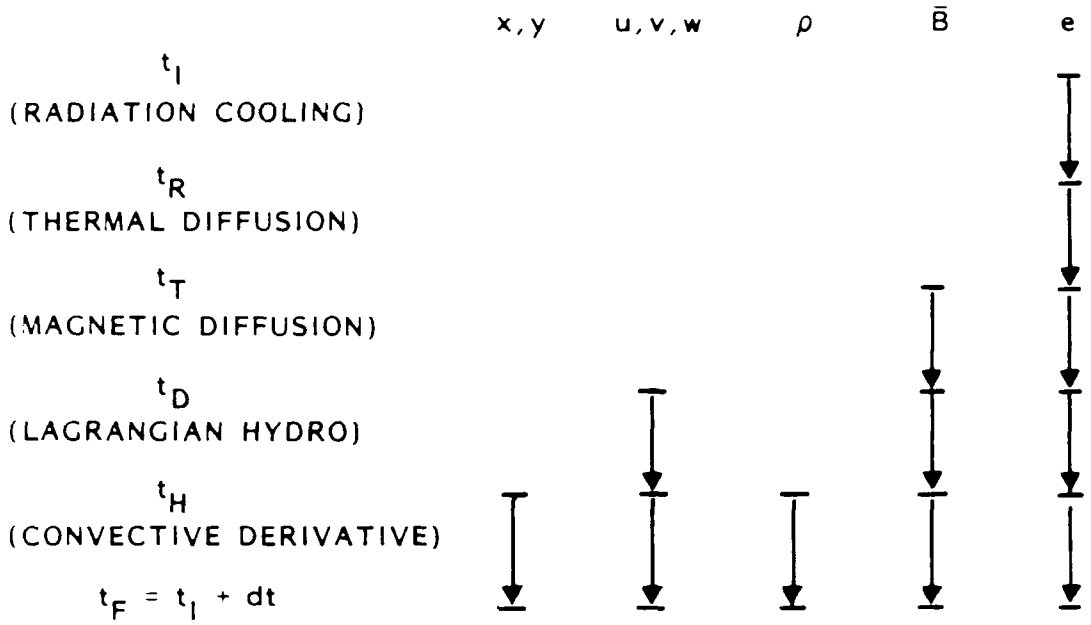
In the subsections that follow, a variable superscripted by I, R, T, D, H, or F is to be understood as having remained unchanged since t_I, t_R, t_T, t_D, t_H, or t_F respectively. Thus, $\overline{v}^H$ will mean the hydrodynamic velocity after the current time-step's advance by the hydromagnetic process, and before that of the convective process.

Since our object here is to describe the time differencing, we will use the continuous vector field differentiation symbols to write the equations. The details of the spatial discretization and differencing will be discussed in Section 7.

6.3 RADIATION COOLING

Radiation cooling is the first physical process in the time split computational cycle. It advances the internal energy from time t_I to t_R by

$$e^R = e^I + dt\, \kappa^I (T^I)^4 \tag{83}$$



AMRC-R-874

Figure 7. The time-split advance of the fundamental variables.

thus it is explicitly time differenced. No time step control is used, however. To prevent the coding from producing negative energies, the cooling rate is limited so that the no more than 90 percent of the internal energy can be removed in one time-step.

6.4 THERMAL DIFFUSION

The thermal diffusion is implicitly time differenced. The computation begins with the determination of a local time centering parameter β. It is computed using

$$\beta = \begin{cases} \beta_{min} & \text{if } C_T > \frac{1}{2\gamma} \left(\beta_{min} + \frac{3}{2}\right) \\ 2C_T - \frac{3}{2} & \\ \beta_{max} & \text{if } C_T < \frac{1}{2\gamma} \left(\beta_{max} + \frac{3}{2}\right) \end{cases} \tag{84}$$

where

$$C_T = \frac{\|k_e\| dt}{\rho C_v (dL)^2} \tag{85}$$

is the thermal Courant number, $\beta_{max} > \beta_{min}$ are control parameters in the range $[0,1]$, and γ is a control parameter greater than 1.

The temperature is then advanced from t_I to t_T by iteratively solving

$$T^T = T^I - \frac{dt}{\rho^I C_v^I} \left[\beta \vec{\nabla} \cdot \left(\bar{\bar{k}}_e^I \vec{\nabla} T^T\right) + (1 - \beta) \vec{\nabla} \cdot \left(\bar{\bar{k}}_e^I \vec{\nabla} T^I\right)\right] \tag{86}$$

for T^T. Thus, for $\beta_{max} = 1.$ and $\beta_{min} = 0.$ the differencing is fully explicit in cells where $C_T < 0.75/\gamma$ and fully implicit where $C_T > 1.25/\gamma$. For C_T between these values, the differencing is of mixed type which, nonetheless, requires iteration. The conductivity tensor is held fixed

during the iteration, even though it may be a strong function of temperature. The iteration is carried out by a multigrid procedure which is described in detail in Section 10. After T^T converges to some desired accuracy, the internal energy ismodified explicitly using

$$e^T = e^I + (T^T - T^I)/(\frac{dT}{de})^I \tag{87}$$

6.5 MAGNETIC DIFFUSION

The third physical process effected in the main cycle is that of resistive diffusion of the magnetic field. This is the second of three processes in the code which are implicitly time differenced. This computation begins with the determination of a local time centering parameter, again using Equation 84 but with C_T replaced by C_M, the magnetic diffusion Courant number, given by

$$C_M = \frac{\eta dt}{(dL)^2} \tag{88}$$

The magnetic field is then advanced from t_I to t_D by iteratively solving

$$\overline{B}^D = \overline{B}^I - dt\ [\beta\overline{\nabla} \times (\eta^I\overline{\nabla} \times \overline{B}^D) + (1-\beta)\overline{\nabla} \times (\eta^I\overline{\nabla} \times B^I)] \tag{89}$$

for B^D. Thus, for $\beta_{max} = 1.$ and $\beta_{min} = 0.$ the differencing is fully explicit in cells where C_M is less than $0.75/\gamma$ and fully implicit where it is greater than $1.25/\gamma$. For values in between the differencing is of a mixed type which, nonetheless, requires iteration. Both the classical and anomalous resistivity remain fixed during the iteration. After the solution is obtained to some desired accuracy, the joule heating is applied to the fluid using

$$e^D = e^T + dt\ (\overline{\nabla} \times \overline{B}^D) \cdot (\eta^I \nabla \times B^D)/\rho^I \tag{90}$$

6.6 LAGRANGIAN HYDRODYNAMICS

This is the other major physical process of the main cycle which is implicitly time differenced. The computation begins by determining the values of the components of the artificial viscosity tensor $\bar{\bar{\sigma}}$ from values of the velocity and density at time t_I. Details of this are reserved for Section 7. The velocity, density, and pressure are then advanced from time t_I to time t_H, and the magnetic field from time t_D to time t_H by iteratively solving

$$\bar{v}^H = \bar{v}^I + \frac{dt}{\rho^I} \bar{\nabla} \cdot \bar{\bar{S}}$$

$$S_{\alpha\beta} = \left(- p^H - \frac{1}{2} B_\gamma^H B_\gamma^H \right) \delta_{\alpha\beta} + B_\alpha^H B_\beta^H + \sigma_{\alpha\beta}{}^I \qquad \alpha, \beta \epsilon \{1,2,3\}$$

$$\rho^H = \rho^I - dt\ \rho^H \bar{\nabla} \cdot \bar{v}^H$$

$$p^H = p^I + \left(\frac{\partial P}{\partial \rho} \Big|_s \right)^I (\rho^H - \rho^I)$$

$$\bar{B}^H = \bar{B}^I - dt\ (\bar{B}^H \bar{\nabla} \cdot \bar{v}^H - \bar{B}^H \cdot \bar{\nabla} \bar{v}^H) \tag{91}$$

Here, $\partial P/\partial \rho \big|_s$ is the square of the adiabatic sound speed. After these are approximately solved to a satisfactory tolerance, the internal energy is advanced to t_H by the appropriate pressure heating or cooling using

$$e^H = e^D + \frac{dt}{\rho^I} \left(- p^H \delta_{\alpha\beta} + \sigma_{\alpha\beta}^I \right) E_{\alpha\beta}^H \qquad \alpha, \beta \epsilon \{1,2\}$$

$$\bar{\bar{E}}_{\alpha\beta}^H = \begin{bmatrix} \dfrac{\partial v_r^H}{\partial r} & \dfrac{1}{2} \left(\dfrac{\partial v_r^H}{\partial z} + \dfrac{\partial v_z^H}{\partial r} \right) \\ \dfrac{1}{2} \left(\dfrac{\partial v_r^H}{\partial z} + \dfrac{\partial v_z^H}{\partial r} \right) & \dfrac{\partial v_z^H}{\partial r} \end{bmatrix} \tag{92}$$

6.7 CONVECTIVE TRANSPORT

All that remains of the full model are the terms due to material derivative, i.e.,

$$\frac{d\phi}{dt} = \frac{\partial \phi_e}{\partial t} + \bar{v} \cdot \bar{\nabla}\phi_e = \frac{\partial \phi_a}{\partial t} + (\bar{v} - \bar{v}_c) \cdot \nabla \phi_a = 0 \tag{93}$$

for $\phi = \rho, e, \bar{v}, \bar{B}$. These are the same as Equations 71 in Section 5, except for the substitution of

$$\frac{\partial \rho}{\partial t} + \bar{\nabla} \cdot (\rho \bar{v}) = 0 \tag{94}$$

for

$$\frac{\partial \rho}{\partial t} + \bar{v} \cdot \bar{\nabla}\rho = 0 \tag{95}$$

Since Equations 71 are shown in Section 5 to be equivalent to Equations 72, they are time differenced instead. The additional $\rho \bar{\nabla} \cdot \bar{v}$ of Equation 94 is accounted for by ignoring ρ^H and using ρ^I in a single place, since Equation 91 shows that to be precisely the difference between them. The time differenced forms are

$$\int_{R^F} \rho^F dV = \int_{R^I} \rho^I dV + dt \int_{\partial R^F} \rho^H (\bar{v}_c - \bar{v}^H) \cdot \bar{n}\, dA$$

$$\int_{R^F} \rho^F e^F dV = \int_{R^I} \rho^I e^H dV + dt \int_{\partial R^F} \rho^H e^H (\bar{v}_c - \bar{v}^H) \cdot \bar{n}\, dA$$

$$\int_{R^F} \rho^F \bar{v}^F dV = \int_{R^I} \rho^I \bar{v}^H dV + dt \int_{\partial R^F} \rho^H \bar{v}^H (\bar{v}_c - \bar{v}^H) \cdot \bar{n}\, dA$$

$$\int_{R^F} \rho^F \bar{B}^F dV = \int_{R^I} \rho^I \bar{B}^H dV + dt \int_{\partial R^F} \rho^H \bar{B}^H (\bar{v}_c - \bar{v}^H) \cdot \bar{n}\, dA \tag{96}$$

where $\bar{v}_c$ is the velocity of the coordinate system, and R^F and R^I are the region at time to t_F and t_I, respectively.

7. SPATIAL DISCRETIZATION, CENTERING, AND DIFFERENCING

There is a not a lot of a detailed numerical analytical nature, such as estimates of accuracy or stability, that requires discussion or that will be illuminating. However, the basis for the numerical analysis must be detailed. This includes the spatial discretization, the centering of the discretized physical variables, and the spatial differencing of all operators appearing in the equations which describe the physical processes given in Section 6. The boundary condition implementation technique and the iterative solution procedures for the implicitly time-stepped processes will also be described here.

7.1 SPATIAL DISCRETIZATION

The description of the class of admissible geometries is the starting point for the discretization of the domain. The problem domain, a block-like complex, is the image of a block complex under the action of a diffeomorphism. If the blocks of that complex are discretized in the most obvious way, a discretization of the block-like complex results from the diffeomorphism. So $B_{\underline{a},\underline{b}}$ is replaced with the collection of integer grid-points it contains

$$B_l = \{(i,j) \in Z^2 \mid 1 \leq i \leq I_l,\ 1 \leq j \leq J_l\},\ l = 1,\ldots,L \qquad (97)$$

Since the location of the domain of the diffeomorphism is immaterial, each block is taken to have lower-left corner at (1,1). As the continuous coordinate system is described by the diffeomorphisms

$$(X_l(n, \xi, t)\ ,\ Y_l(n, \xi, t)),\ l = 1,\ldots,L \qquad (98)$$

as in Section 4, then the discrete coordinate system is described by

$$\{(x^l_{ij},\ y^l_{ij}) \mid 1 \le i \le I_l,\ 1 \le j \le J_l,\ l = 1,\ldots,L\} \tag{99}$$

a finite collection, indexed by l, of two dimensional arrays of points in the computational plane. These arrays of points are referred to as the grid, or the mesh, in which context they are thought of as being connected to their nearest neighbors which have the same value of i, or of j, as shown in Figure 8. The four points

$$\begin{aligned} &(x^l_{ij},\ y^l_{ij}) \quad (x^l_{i+1,\,j},\ y^l_{i+1,\,j}) \\ &(x^l_{i,\,j+1},\ y^l_{i,\,j+1}) \quad (x^l_{i+1,\,j+1},\ y^l_{i+1,\,j+1}) \end{aligned} \tag{100}$$

which lie in general position, and the four lines of constant i or constant j which connect them form the cell indexed by (i,j). The location

$$\begin{aligned} (x^l_{cij},\ y^l_{cij}) = \frac{1}{4}\big[&(x^l_{ij},\ y^l_{ij}) + (x^l_{i+1,j}\ y^l_{i+1,j}) \\ &+ (x^l_{i,j+1},\ y^l_{i,j+1}) \\ &+ (x^l_{i+1,\,j+1},\ y^l_{i+1,\,j+1})\big] \end{aligned} \tag{101}$$

is called the cell center, and points of the arrays are known as vertices. Such a mesh is often referred to as an arbitrary quadrilateral mesh, for obvious reasons.

All of the variables in the code which have spatial dependence are stored as functions of l and (i,j), so that the density, for example, is known as a function of (x,y) only through the parametric description given by Equation 99 and the collection of arrays

$$\{\rho^l_{ij}\} \tag{102}$$

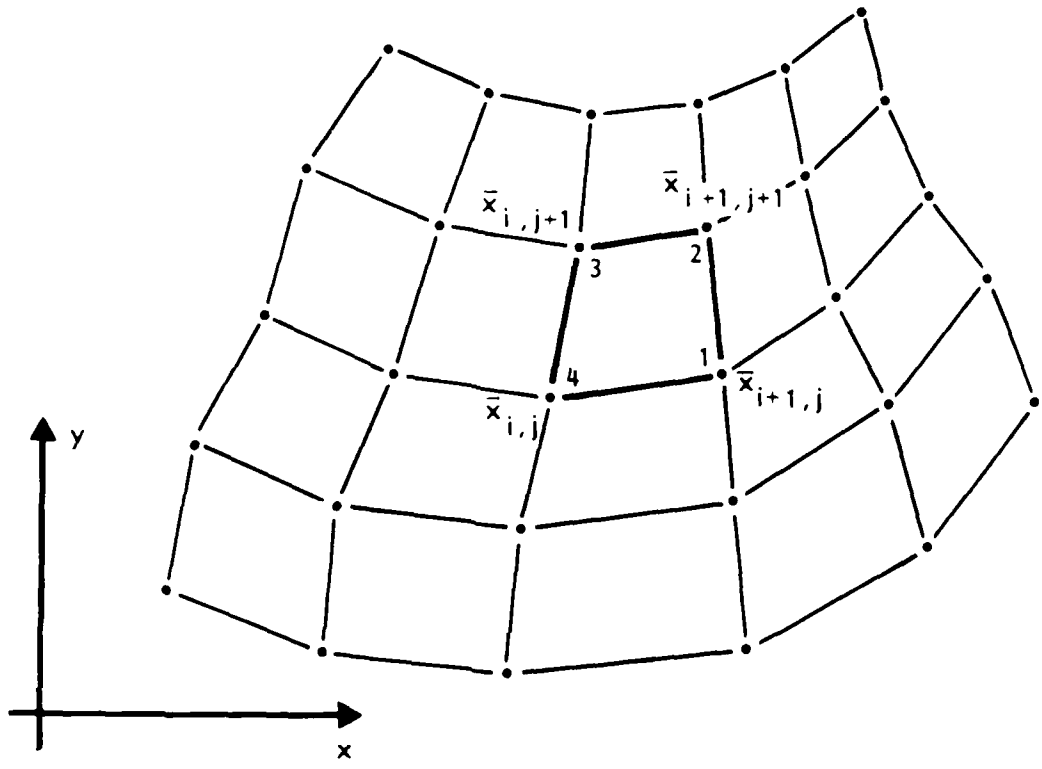


Figure 8. Computational grid indexing and corner numbering.

7.2 SPATIAL CENTERING

For the purpose of deriving difference formulas, most spatially varying quantities are thought of as having values located at either cell centers or vertices. The fundamental variables of the code are thus split into two classes:

Vertex Centered -- position, velocity

Cell Centered -- density, internal energy, magnetic field

There are many other space dependent quantities derived from these, and they are usually quite easily identified as belonging to one or the other class by the following simple rule: difference operators map each class to the other. Thus, the current density is a vertex centered quantity, since $J = \overline{\nabla} \times \overline{B}$. When it is necessary to call attention to their centering, cell quantities will be indicated by a superscript "c", and vertex quantities by a superscript "v".

Unfortunately, a third class of variable exists: the edge centered quantity. The convective fluxes of mass, momentum, internal energy, and magnetic field fall into this class; they are all derived quantities. When it necessary to call attention to such a quantity's centering it will be superscripted with an "e".

7.3 FINITE VOLUME DIFFERENCING

In general, the finite volume approach is used to derive the differencing in the code. This approach can best be understood by considering the example of $\overline{\nabla} \times (\ )$. One form of Stokes Theorem that involves this operator is

$$\int_R \overline{\nabla} \times \overline{F} \, dV = \int_{\partial R} \overline{n} \times \overline{F} \, dA \qquad (103)$$

Here R is region of 3-space and ∂R is its bounding surface. If R is taken to be the small, discrete volume in 3-space described by a single cell in the computational grid, then Equation 103 becomes

$$(\bar{\nabla} \times \bar{F})^{1}_{ij} \, dV^{1}_{ij} = \sum_{\text{faces}} (\bar{n} \times \bar{F})^{1}_{ij} \, dA^{1}_{ij} \tag{104}$$

where it has been assumed that $\bar{\nabla} \times \bar{F}$ and $\bar{n} \times \bar{F}$ are uniform over the volume and its faces, respectively. The quantity dV^{1}_{ij} here is the cell volume and dA^{1}_{ij}, the face area. If values for $\bar{n} \times \bar{F}$ are specified for each of the faces of the cell, then Equation 104 can be used to define $\bar{\nabla} \times \bar{F}$. The differences are hidden in the direction of the normals in the sum on the right hand side; the normal on one face is opposed in sense by the normal on the opposite face. In what follows, the indices $(\ )^{1}_{ij}$ will be omitted.

The principal advantage of this scheme of differencing is that conservation laws written in terms of the quantities in the vector integral theorems are usually well respected by the differencing. In fact, for the example above, all that is required to insure that Equation 104 is satisfied exactly for R a region composed entirely of whole cells is that the quantity $\bar{n} \times \bar{F}$ be computed the same way, except for its sign, for a particular face regardless of the cell with which it is associated. This is worthy of more explication. First note that

$$\sum_{\text{cells}} (\nabla \times F) \, dV = \sum_{\text{cells}} \sum_{\text{faces}} (n \times F) \, dA \tag{105}$$

Since, by the above assumption, two terms $(n \times F) \, dA$ appear within the double sum for each cell interface, each with opposing sign, all but the contributions from the exterior faces must cancel. Hence it is true, exactly, without approximation, that

$$\sum_{\text{cells}} (\nabla \times F) \, dV = \sum_{\substack{\text{exterior} \\ \text{faces}}} (n \times F) \, dA \tag{106}$$

This is the best possible statement of Equation 103 for the discretized data available.

There are two subtleties in the application of the finite volume differencing to any particular case, however. The first is due to the possible cylindrical symmetry. The second is due to the different centering types of the data. There is also a bit of unpleasantry: the requirement to specify $(n \times F)\, dA$ precisely in each case. Each of these shall be dealt with in turn.

In cylindrical symmetry, though all vector components satisfy

$$\frac{\partial}{\partial \theta} F_i = 0 \tag{107}$$

the standard basis $(\hat{r}, \hat{\theta}, \hat{z})$ satisfies

$$\frac{\partial \hat{r}}{\partial \theta} = \hat{\theta}$$

$$\frac{\partial \hat{\theta}}{\partial \theta} = - \hat{r} \tag{108}$$

If the three-dimensional discrete volume associated with a two-dimensional cell in (r,z) space is assumed to extend a small increment $d\theta$ in the $\hat{\theta}$ direction, as shown in Figure 9, then Equation 104 becomes

$$\bar{\nabla} \times \bar{F} = \frac{1}{\left(\frac{dV}{d\theta}\right)} \left\{ \frac{1}{d\theta} \left(\hat{\theta}\,(\theta + d\theta) - \hat{\theta}\,(\theta)\right) \times \bar{F}\, da \right.$$

$$\left. + \sum_{\substack{\text{cell} \\ \text{edges}}} \bar{n} \times \bar{F}\, r dl \right\} \tag{109}$$

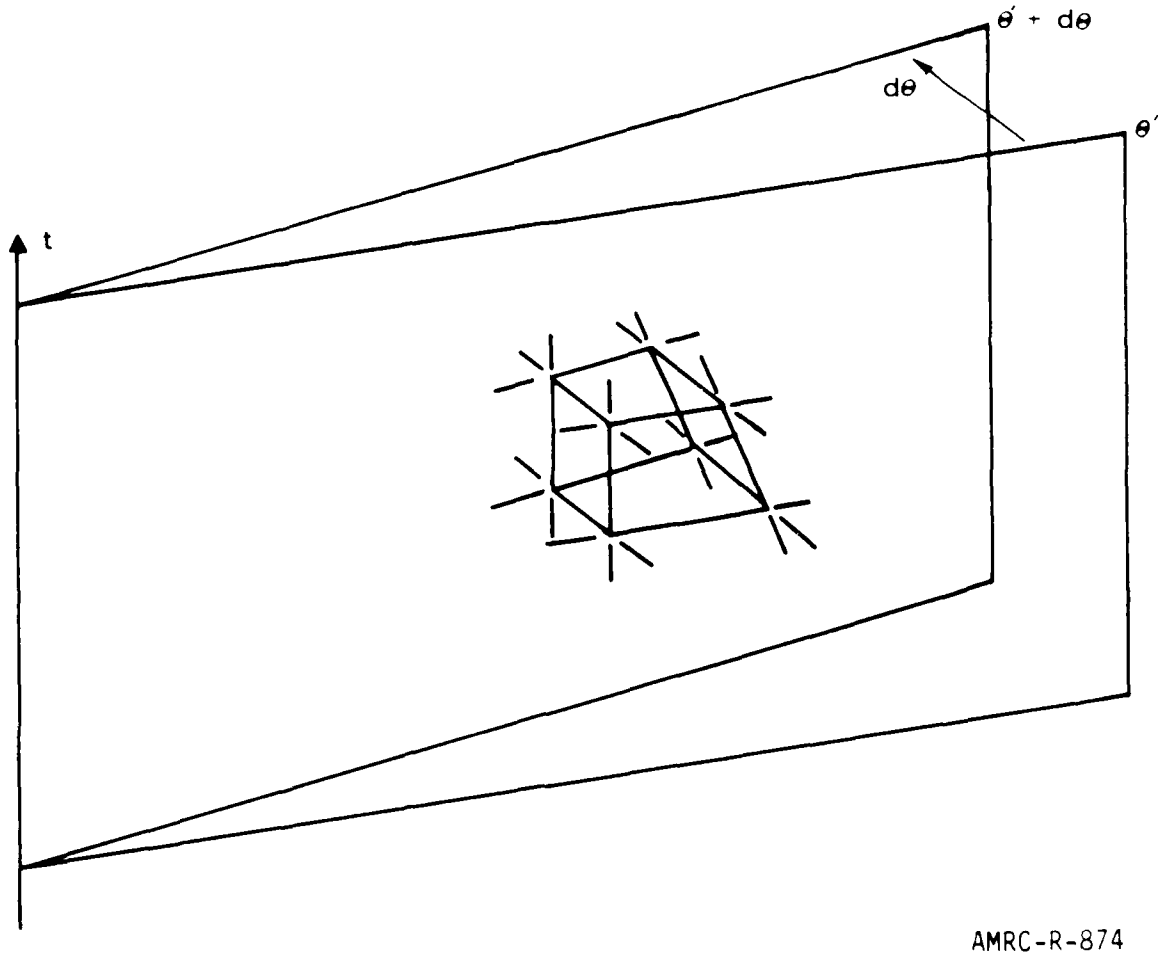


Figure 9. Finite volume for cell centered differences of vertex quantities.

since $\bar{F}$ is independent of ϑ. Here da is the area of the cell in the computational plane, and the sum extends over the four faces obtained by extending the cell edges normal to the computational plane by the increment $rd\theta$. Taking the limit as $d\theta$ goes to zero and using Equation 108 we obtain

$$\bar{\nabla} \times \bar{F} = \frac{1}{\left(\frac{dV}{d\theta}\right)} \left\{ -\hat{r} \times \bar{F}\, da + \sum_{\substack{\text{cell} \\ \text{edges}}} \bar{n} \times \bar{F}\, rdl \right\} \tag{110}$$

Thus the difference formula for $\bar{\nabla} \times \bar{F}$ contains terms in cylindrical symmetry that are absent in planar symmetry. In the code planar symmetry is easily implemented by setting the quantity da to zero and the radius r in the summation in Equation 110 to one.

It might seem that all occurrences of the curl operator could be differenced alike; sadly, this is not the case. There are two distinct difference forms for it in use in the code:

cell centered curl of vertex quantity
vertex curl of a cell centered quantity

and this problem extends to other operators as well. In large measure the cases differ only in the choice of the volume over which the technique is applied. The finite volume in Figure 9 is the one appropriate for computing cell centered differences of vertex centered quantities. Figure 10 shows a region in the computational plane which, when extended by $rd\theta$ in the $\hat{\theta}$ direction, becomes the finite volume that is used to compute vertex centered differences of cell centered quantities. The region consists of four parallelograms formed by the vertex (i,j) and the midpoints of adjacent pairs of edges leaving it. Since this is the most common form of differencing in the code, some geometric quantities associated with this finite volume are computed once per cycle and saved for reuse to avoid recomputation.

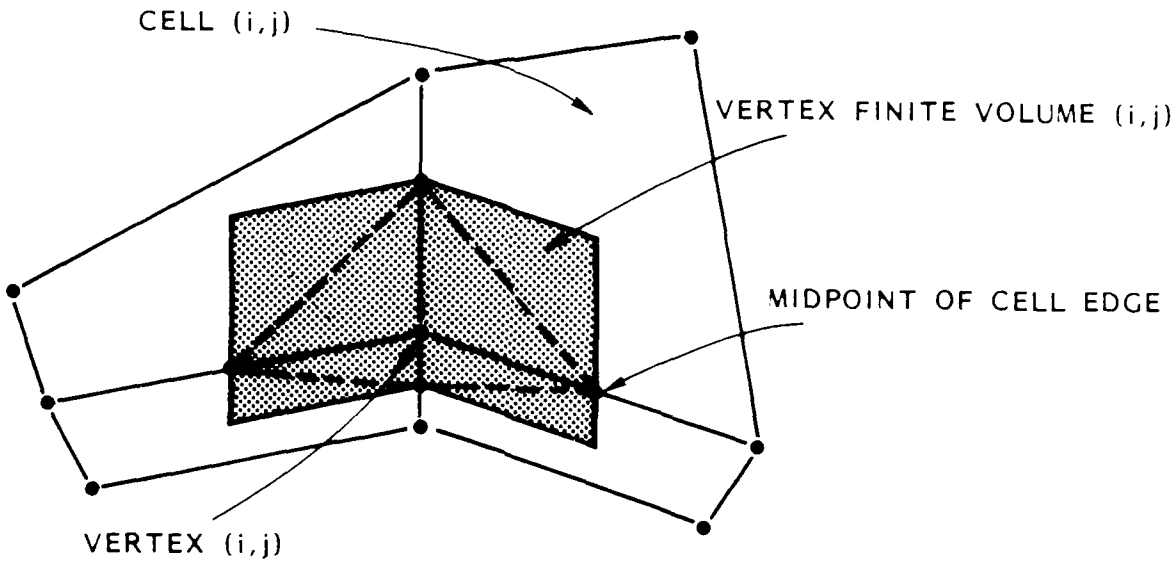
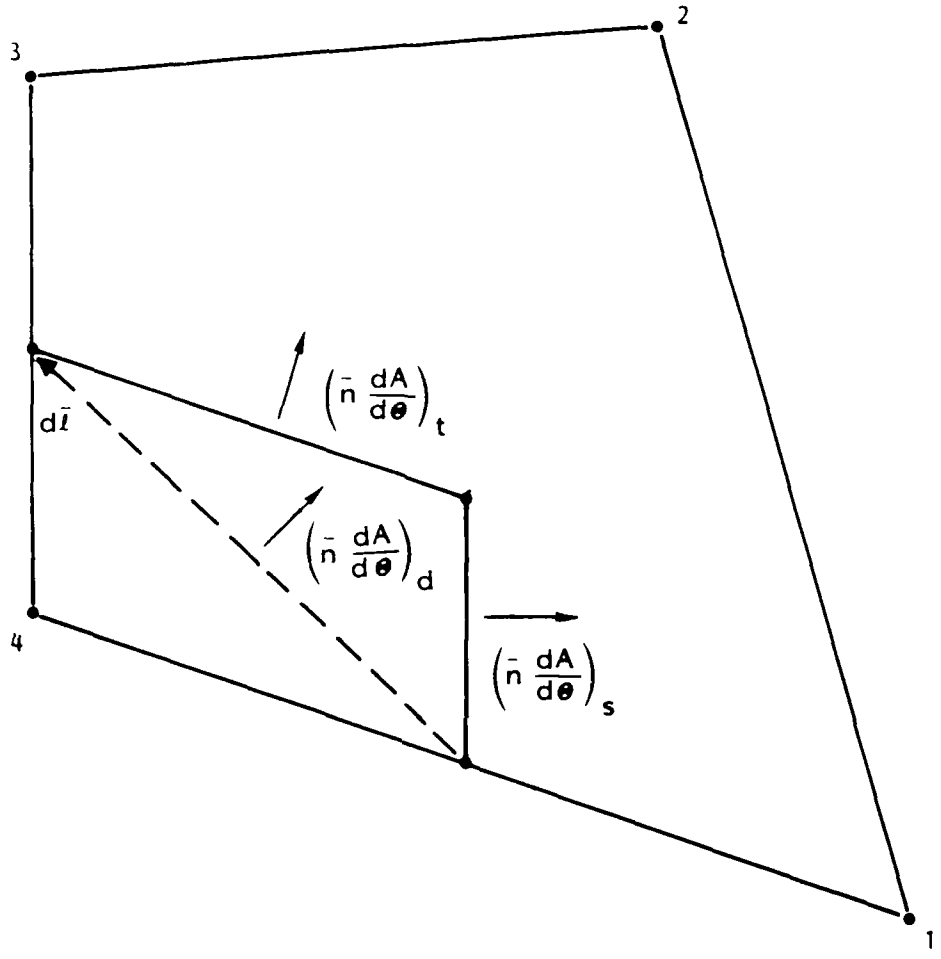


Figure 10. Projection on computational plane of finite volume for vertex centered differences of cell centered quantities.



AMRC-R-874

Figure 11. Cell corner area-weighted normals for vertex centered differencing of cell centered data.

7.4 STORED GEOMETRIC COEFFICIENTS

Those geometric quantities are most easily understood in the context of the derivation of the vertex centered gradient of some cell centered scalar quantity ϕ. Applying

$$\int_R \nabla\phi \, dV = \int_{\partial R} \phi \, \bar{n} \, dA \tag{111}$$

where $\bar{n}$ is an outward pointing vector field, to the finite volume described above yields

$$(\nabla\phi)^v = \frac{1}{(\frac{dV}{d\theta})^v} \left\{ -\hat{r} \sum_{\substack{\text{adjacent} \\ \text{cells}}} \phi^c \, da^c + \sum_{\text{edges}} \phi^c \, \bar{n} \, \frac{dA}{d\theta} \right\} \tag{112}$$

Here the first sum is over the four shaded parallelograms and the second is over the eight edges of the shaded figure of Figure 10.

The second sum is collapsed to four terms and converted to a sum over adjacent cells by assuming that within each cell ϕ is constant. The two terms

$$\phi \left(\bar{n} \frac{dA}{d\theta}\right)_t + \phi \left(\bar{n} \frac{dA}{d\theta}\right)_s \tag{113}$$

within the sum which derive from a single cell, shown in Figure 11, are replaced with the single term

$$\phi \left(n \frac{dA}{d\theta}\right)_d \tag{114}$$

where the subscripts t, s, and d stand for top, side, and diagonal as illustrated in the figure. This is equivalent to assuming

$$\left(\bar{n}\,\frac{dA}{d\theta}\right)_t + \left(\bar{n}\,\frac{dA}{d\theta}\right)_s = \left(\bar{n}\,\frac{dA}{d\theta}\right)_d \tag{115}$$

which is not precisely true in cylindrical symmetry, since it ignores the variation in dA with r within the cell. This does not lead to disastrous results since we are using the finite volume approach. The boundary quantities need only be estimated, not computed exactly, since, so long as they are all computed consistantly, the integral conservation laws remain valid. The same radius is used for each of the four directed vertex face areas in one cell; that radius is the centroid of the cell area. These areas are thus given by

$$\left(\bar{n}\,\frac{dA}{d\theta}\right)_k^c = r^c\, R\, d\bar{l}_k \qquad k = 1,\ldots,4 \tag{116}$$

where r^c represents the cell centroid, R is a rotation by $\pi/_2$ radians, $d\bar{l}$ is the diagonal of the corner parallelogram as indicated in Figure 11 and the corner index k is as shown in Figures 8 and 11.

The four vector quantities in Equation 116, and the quantity

$$\left(\frac{dV}{d\theta}\right)^v \tag{117}$$

referred to as the vertex volume, and the areas da_k^c for k = 1,...,4 of parallelograms such as the one in Figure 11, completely specify the vertex differences of cell centered quantities. They are computed once per cycle and stored for reuse. The corner areas da_k^c and the directed vertex areas $\left(\bar{n}\,\frac{dA}{d\theta}\right)_k^c$ are cell quantities, as indicated by the superscript c, but are each associated with a particular vertex of the cell, and are numbered as the vertices are numbered in Figure 8. In addition to these, the four volumes

$$\left(\frac{dV}{d\theta}\right)_k^c = r^c da_k^c \qquad k = 1,\ldots,4 \tag{118}$$

referred to as the corner volumes, and the reciprocal cell volume

$$\frac{1}{\left(\frac{dV}{d\theta}\right)^c} = \left(\sum_{k=1}^{4} \left(\frac{dV}{d\theta}\right)_k^c\right)^{-1} \tag{119}$$

are stored for use in other differencing and averaging. Note that the vertex volume is defined as

$$\left(\frac{dV}{d\theta}\right)^v = \sum_{\substack{\text{adjacent} \\ \text{cells}}} \left(\frac{dV}{d\theta}\right)_{k(c)}^c \tag{120}$$

where the notation k(c) indicates that the appropriate corner number is to be chosen within each cell so that the corner referred to is adjacent to the vertex at which the volume is being computed. One advantage of computing these geometric quantities in only one place in the code is that they can all be easily changed in a consistent way if that should be necessary. That has been done on at least one occasion.

With the above definitions, the difference form of the vertex centered gradient of a cell centered scalar quantity ϕ may be written as

$$\nabla\phi^v = \frac{1}{\left(\frac{dV}{d\theta}\right)^v} \left\{ \sum_{\substack{\text{adjacent} \\ \text{cells}}} \phi^c \left[-\hat{r}\, da_{k(c)}^c + \left(\bar{n}\, \frac{dA}{d\theta}\right)_{k(c)}^c\right] \right\} \tag{121}$$

8. DIFFERENCE OPERATORS BY PHYSICAL PROCESS

The remainder of this section will follow the organization of the main computational cycle, dealing with each type of differencing for each operator where it first makes an appearance. The precise specification of the boundary quantities for non-vertex-centered difference operators will also be covered along the way, since they depend on the centering type of the data to which the operator applies.

8.1 RADIATION COOLING

There is no differential operator in Equation 83 which governs the radiation cooling model. That is, spatial derivatives play no role in energy loss by free-free emission.

8.2 THERMAL DIFFUSION

There are two differential operators, the divergence and the gradient, in the thermal diffusion model of Equation 86. The divergence must produce a cell centered quantity, the thermal heating rate, and the gradient is of the temperature, a cell centered quantity. The centering of the gradient is arbitrary, so it may be taken to be a vertex quantity.

Since the gradient of the temperature is a vertex difference of a cell centered quantity, it is computed using Equation 121 with $\phi = T$ and the geometric quantities defined above. The thermal flux, also a vertex quantity, is computed by multiplying by the effective thermal conductivity, so that

$$\overline{F} = \bar{\bar{k}}\ \overline{\nabla}\ T \tag{122}$$

The divergence of the flux is then computed to determine the cell centered thermal heating rate. Because the thermal flux is a vertex centered quantity and the thermal heating rate is cell centered, the geometric quantities do not apply exactly as derived. However, the boundary integral areas for vertex centered differences are related to those for cell centered differences. This relationship is made clear in Figure 12. The assumption there is that the areas of the surfaces associated with the cell edges do not vary with radius. To correct for this error, additional terms proportional to the cell corner areas $da^c_{k(v)}$ are included in the difference formula. Thus, the divergence of thermal flux is computed using

$$(\nabla \cdot \bar{F})^c = \frac{1}{\left(\frac{dV}{d\theta}\right)^c} \sum_{\text{vertices}} \left[-\hat{r}\, da^c_{k(v)} - \left(\bar{n}\, \frac{dA}{d\theta}\right)^c_{k(v)}\right] \cdot \bar{F}^v \qquad (123)$$

8.3 MAGNETIC DIFFUSION

The only differential operator in Equation 89, which governs diffusion of magnetic field, is the curl. However, two centering types are required. As mentioned above, since $\bar{B}$ is cell centered, $\bar{J}$ is taken to be vertex centered. Hence, the difference operator in

$$\bar{J} = \bar{\nabla} \times \bar{B} \qquad (124)$$

is a vertex centered difference of a cell centered quantity, while that in

$$\frac{\partial \bar{B}}{\partial t} = \bar{\nabla} \times (\bar{\bar{n}} \cdot \bar{J}) \qquad (125)$$

is a cell centered difference of vertex data.

The derivation of the difference formula for the cell centered curl of a vertex centered quantity was begun above, in the introductory discussion on finite volume differencing. All that remains to complete the derivation is to specify the quantities on the right hand side of Equation 110, repeated here with $\bar{E} = \bar{\bar{n}} \cdot \bar{J}$ substitued for $\bar{F}$

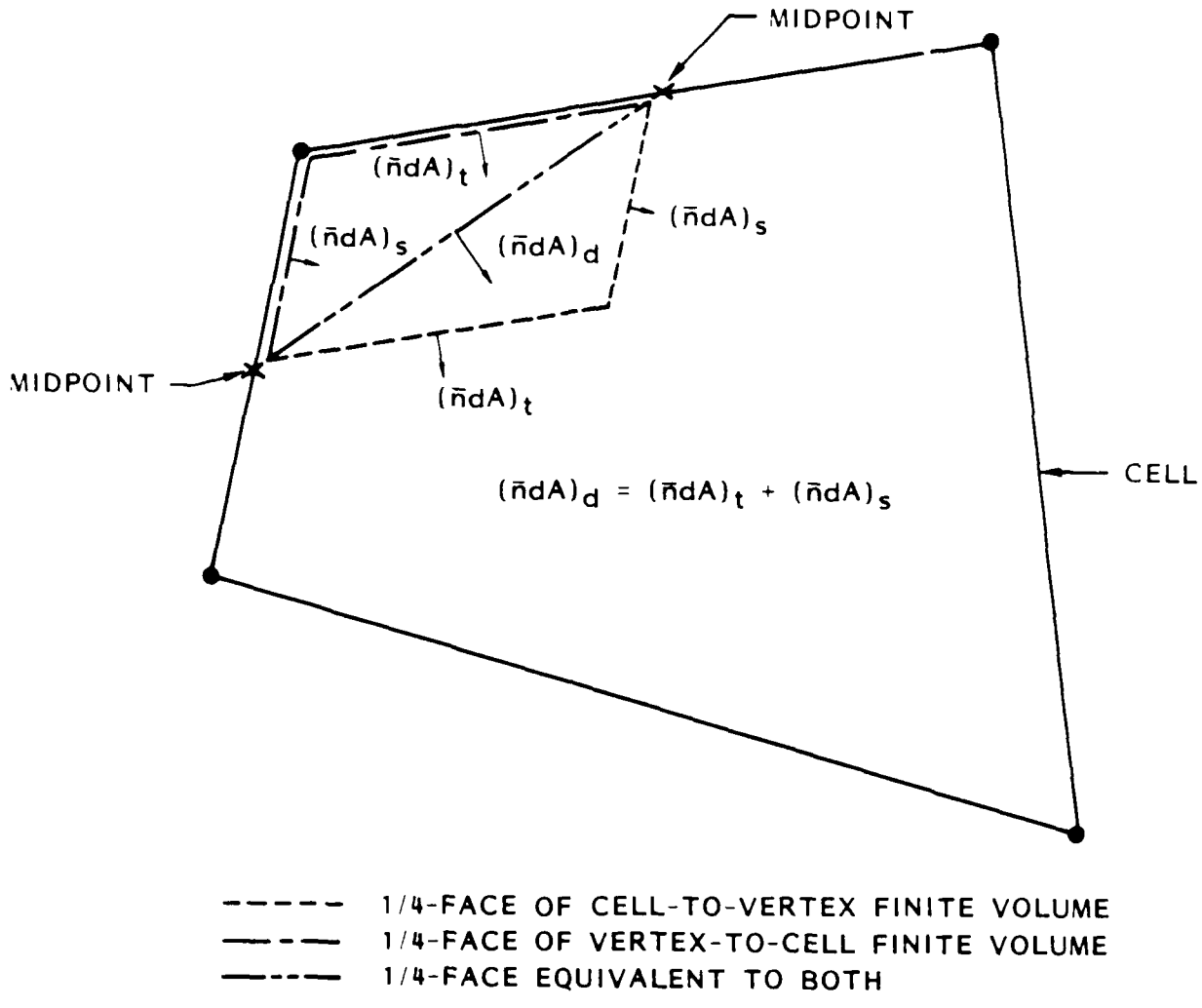


Figure 12. Equivalence of cell-to-vertex and vertex-to-cell area weighted normals.

$$\bar{\nabla} \times \bar{E} = \frac{1}{\left(\frac{dV}{d\theta}\right)} \left\{ -\hat{r} \times \bar{E}\, da + \sum_{\substack{\text{cell} \\ \text{edges}}} \bar{n} \times \bar{E}\, r dl \right\} \tag{126}$$

for easy reference. The volume on the right hand side needs to be the volume of the cell per radian, and hence the reciprocal cell volume, one of the saved geometric quantities, is used there. Since $\bar{E}$ is vertex centered, the first term in the braces must be expanded into an average over the four vertices of the cell

$$\hat{r} \times (\bar{E} da)^c = \hat{r} \times \left(\sum_{\text{vertices}} \bar{E}^{v(k)}\, da_k^c \right) \tag{127}$$

The da_k^c here are the cell corner areas, which are also among the previously defined geometric quantities, and the notation v(k) indicates that the $(\quad)^{\ell}_{i,j}$ indices of $\bar{E}$ are chosen to match the corner index k as in Figure 8. The terms of the sum which remains are first rearranged as

$$(r\bar{E})^e \times (\bar{n} dl)^e \tag{128}$$

The first factor is defined by

$$(r\bar{E})^e = \left(\frac{1}{3}r + \frac{1}{6}r'\right) \bar{E} + \left(\frac{1}{6}r + \frac{1}{3}r'\right) \bar{E}' \tag{129}$$

where, as before, the primed and unprimed quantities refer to the two vertices which are the ends of the appropriate edge. The second factor is taken to be

$$(\bar{n} dl)^e = R d\bar{l} \tag{130}$$

in which R is the $\pi/2$ rotation operator, and $d\bar{l}$ is the edge, oriented so that the cell boundary is traversed counterclockwise.

The vertex centered curl of cell centered data is derived using the finite volume described by Figure 10. It is given by

$$(\bar{\nabla} \times \bar{B})^V = \frac{1}{(\frac{dV}{d\theta})^V} \left\{ \sum_{\substack{\text{adjacent} \\ \text{cells}}} [-\hat{r}\, da^c_{k(c)} + (\bar{n} \frac{dA}{d\theta})^c_{k(c)}] \times \bar{B}^c \right\} \quad (131)$$

where all the geometric quantities have already been defined.

8.4 LAGRANGIAN HYDRODYNAMICS

Here the differential operators in the equations of advance are the divergence and the covariant derivative, that is

$$\bar{\nabla} \cdot (\quad) \quad \text{and} \quad (\quad) \cdot \bar{\nabla} (\quad) \quad (132)$$

The divergence is required in two centering types, while the covariant derivative must produce a cell centered result from vertex centered data.

First, the momentum equation requires a vertex centered divergence of a cell centered tensor of rank two. This difference formula can thus be written in terms of the saved geometric quantities. The form is

$$(\bar{\nabla} \cdot \bar{\bar{S}})^V = \frac{1}{(\frac{dV}{d\theta})^V} \sum_{\substack{\text{adjacent} \\ \text{cells}}} [(S_{\theta r}\hat{\theta} - S_{\theta\theta}\hat{r})\, da^c_{k(c)} + (\bar{n} \frac{dA}{d\theta})^c_{k(c)} \cdot \bar{\bar{S}}] \quad (133)$$

which differs from Equation 131 by the substitution of the cross product for the dot product, and since $\bar{\bar{F}}$ is a tensor instead of a vector.

The second occurrence of the divergence here is in the terms representing the effects of the velocity field on the magnetic field and mass density. Both of these require a cell centered divergence of a vertex centered quantity, the velocity. The divergence of the velocity is done precisely the same as the divergence in the thermal transport, that is,

$$(\overline{\nabla} \cdot \overline{v})^C = \frac{1}{\left(\frac{dV}{d\theta}\right)^C} \sum_{\text{vertices}} \left[-\hat{r}\, da^C_{k(v)} - \left(\overline{n}\, \frac{dA}{d\theta}\right)^C_{k(v)}\right] \cdot \overline{v}^v \qquad (134)$$

The covariant derivative required is $\overline{B} \cdot \overline{\nabla v}$, and by a similar argument to that given above, is given by

$$(\overline{B} \cdot \overline{\nabla v})^C = \frac{1}{\left(\frac{dV}{d\theta}\right)^C} \sum_{\text{vertices}} \left[B^C_\theta \left(u^v \hat{\theta} - w^v \hat{r}\right) da^C_{k(v)} - \overline{B}^C \cdot \left(\overline{n}\, \frac{dA}{d\theta}\right)^C_{k(v)}\right] \qquad (135)$$

8.5 CONVECTIVE TRANSPORT

The integral statements of the convective process, Equations 96 of Section 6, can be differenced directly. Application of the first of those integrals, which governs the flux of mass, to a single cell produces

$$\rho^F \left(\frac{dV^F}{d\theta}\right)^C = \rho^I \left(\frac{dV^I}{d\theta}\right)^C + dt \sum_{\text{edges}} \rho^{He} \left(\overline{v}_c - \overline{v}^H\right) \cdot \overline{n}^F \frac{dA^F}{d\theta} \qquad (136)$$

The fractional timesteps of the cell volumes have to be displayed since the grid movement step is combined with the convective step. No terms due to w appear here since the density is independent of θ. When the convection of the vector quantities $\overline{B}$ and $\overline{v}$ is considered below, the turning of the basis vectors will require treatment of such terms. The edge quantities

$$dV^e = \left(\overline{v}_c - \overline{v}^H\right) \cdot \overline{n}\, \frac{dA^F}{d\theta}\, dt \qquad (137)$$

which are the rate of flux of fluid volume across the edges, are taken to be

$$\frac{1}{2}\left[\left(\overline{v}_c - \overline{v}^H\right) r_1 + \left(\overline{v}_c - \overline{v}^H\right)' r_2\right] \cdot R d\overline{T} \qquad (138)$$

and

$$r_1 = \frac{1}{3}(2r + r') \qquad r_2 = \frac{1}{3}(r + 2r') \tag{139}$$

The specification of the edge value of the density determines the order of accuracy and the stability of the difference scheme for its transport. In general, techniques for selecting the edge value are based on donor cell averaging, which is of first order accuracy, and linear interpolation, which is of second order. The order of accuracy referred to here is that relevant to smooth functions, i.e., those for which the relative change from cell to cell is small. In such a circumstance second order differencing is clearly superior. However, if the relative cell to cell change is large, both schemes produce errors. Second order differencing tends to introduce unphysical behavior in the solution while first order differencing tends to spread short scale length phenomena. Since neither scheme can support scale lengths shorter than the mesh size, it is usually felt that first order differencing is superior in these regions. Thus these schemes are often combined in some way, in an effort to increase accuracy where the solution is smooth and decrease unphysical behavior where it is not.

Flux corrected transport, described by Zalesak (Ref. 9), is one approach to this averaging. Another approach is to form a combination of the two schemes, in effect, an average of the averages. If the weights are chosen adaptively, as some function of the computed solution, the combined difference scheme can be made second order accurate where the solution is smooth and first order where it is not. This is the approach followed by Brackbill in MOQUI, and adopted here. Letting the superscripts i and o stand for inside and outside, the edge value of the density is given by

$$\rho^{He} = \frac{1 + \alpha}{2}\rho^{Hi} + \frac{1 - \alpha}{2}\rho^{Ho} \tag{140}$$

where the function α that determines the order of the differencing is

$$\alpha = \beta \frac{dV^e}{V^e} + (1 - \beta)\, s \tag{141}$$

Here, V^e is the edge volume, dV^e is the edge volume flux given in Equation 137, and $s = \text{sign}\,(dV^e)$. Where $\beta = 1$, the edge density ρ^e is determined by linear interpolation between ρ^i and ρ^o. Where $\beta = 0$, it is either ρ^i or ρ^o, depending only on whether the volume flux is into or out of the cell in question, and hence the differencing will be donor cell. For values of β in (0,1) the difference scheme is a linear combination of the two. The control parameter β is determined by

$$\beta = \min\left(\beta_v, \beta_\rho\right) \tag{142}$$

The function

$$\beta_v = 1 - \frac{1}{2}\left(h(g^i) + h(g^o)\right) \tag{143}$$

is a measure of the local gradient of the relative velocity of the grid and the fluid, with h as shown in Figure 12, and g computed as

$$g = \frac{d\left|\,\bar{\nabla}\left(\bar{v}_c - \bar{v}^H\right)\right|}{\left|\,\bar{v}_c - \bar{v}^H\right|} \tag{144}$$

where d is measure of the local grid spacing. The choice

$$g_{max} = 0.1 \qquad h_{min} = 0.0 \tag{145}$$

for the parameters of Figure 13 would cause donor cell differencing to be used wherever the vertex to vertex variation of the relative velocity of the grid and fluid exceeds 10 percent. Of course, the choice

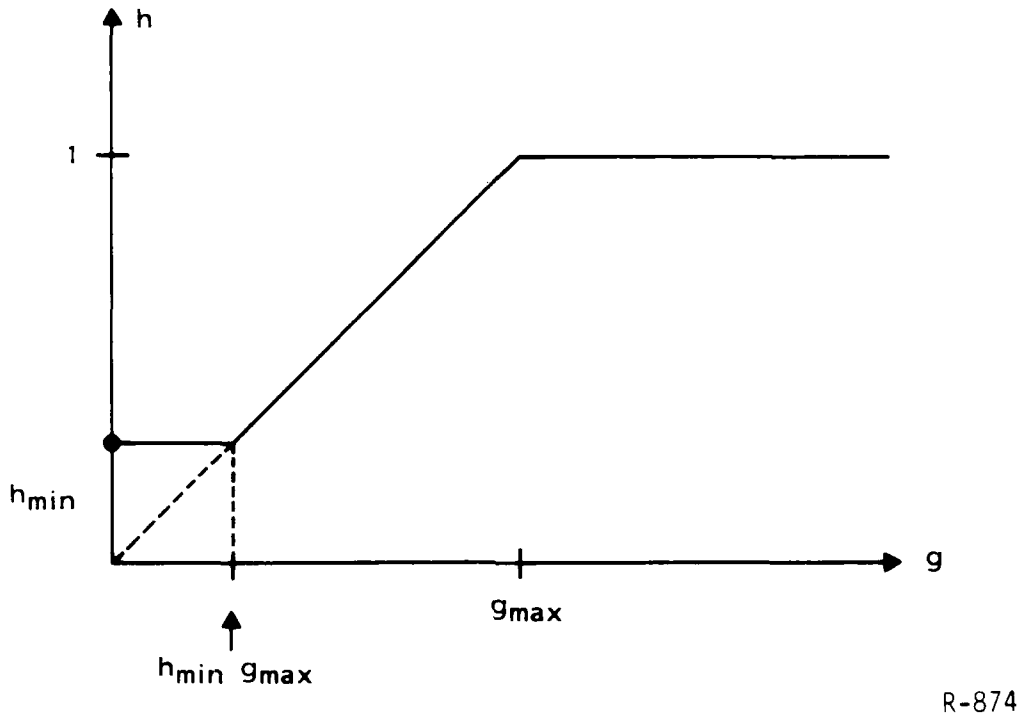


Figure 13. Convective flux differencing control function.

$$h_{min} = 1.0 \tag{146}$$

forces donor cell differencing to be used everywhere. This is the default choice, and the only one which has ever been successfully used for real simulations.

The other control function in Equation 142 is a measure of the density gradient. It is computed as

$$\beta_\rho = \frac{1}{4}\left\{(1+s)\left[(1-s_\rho)+(1+s_\rho)\frac{\rho^i}{\rho^o}\right]\right. + \left.(1-s)\left[(1+s_\rho)+(1-s_\rho)\frac{\rho^o}{\rho^i}\right]\right\} \tag{147}$$

where $s_\rho = \text{sign}(\rho^o/\rho^i - 1)$. The reader is invited to show that this produces 1 if the velocity of the grid relative to the fluid is toward larger density, and $\rho^{small}/\rho^{large}$ if that velocity is toward smaller density. The latter value has the effect of limiting the edge density to $(3/2)\ \rho^{small}$ when the ratio of the densities is very small and the grid is moving into the lower density fluid, rather than the allowing the large value that linear interpolation between ρ^{small} and ρ^{large} would produce.

8.5.1 Homogeneity of the Transport Scheme--It is desirable for the mass transport process to have the following property:

> Homogeneity: For 0 fluid velocity and uniform density, and for any grid velocity, the density after the transport step should be the same as the density before.

Letting ρ_0 be the value of the uniform density, it is clear from Equation 140 that $\rho^{0_{He}} = \rho_0$. Equations 136 and 137 then imply that

$$\rho^F = \rho_0 \left\{\left(\frac{dV^I}{d\theta}\right)^c + \sum_{edges} dV^e\right\} / \left(\frac{dV^F}{d\theta}\right)^c \tag{148}$$

Hence $\rho^F = \rho_0$ if and only if

$$\sum_{\text{edges}} dV^e = \left(\frac{dV^F}{d\theta}\right)^c - \left(\frac{dV^I}{d\theta}\right)^c \tag{149}$$

that is, if only if the estimates of volume flux due to coordinate system motion exactly account for the changes in cell volume due to grid motion. Unfortunately, the volume flux estimates given by the expressions above do not satisfy this. The errors are of two kinds. First, the fluxes of volume to the diagonal neighbor cells, are not included in the estimates. Second, even when the grid moves so that the diagonal fluxes are zero, the edge volume flux estimates are in error due to incorrect radius weighting and improper time centering. The complexity of making the volume fluxes exact is deemed to be a greater penalty than the compensation that would result.

8.5.2 Internal Energy Transport--The other transport equations of Equation 96 can be made homogeneous by the use of mass flux based transport. For example, consider the spatially discretized version of the second of Equations 96, which governs energy transport,

$$\rho^F e^F \left(\frac{dV^F}{d\theta}\right)^c = \rho^I e^H \left(\frac{dV}{d\theta}\right)^c + dt \sum_{\text{edges}} \rho^{He} e^{He} \left(\overline{v}_c - v^H\right) \cdot \overline{n} \frac{dA^F}{d\theta} \tag{150}$$

If the cell mass m^c is defined as

$$m^c = \rho\left(\frac{dV}{d\theta}\right)^c \tag{151}$$

and the edge mass flux dm^e as

$$dm^e = \rho^{He} \left(\overline{v}_c - \overline{v}^H\right) \cdot n^F \frac{dA^F}{d\theta} dt = \rho^{He} dV^e \tag{152}$$

then Equation 150 can be written as

$$e^F m^{Fc} = e^H m^{Ic} + \sum_{\text{edges}} e^e \, dm^e \tag{153}$$

These edge mass fluxes are precisely the same as those computed during the mass transport phase. To emphasize this fact note that Equation 136 can be written

$$m^{Fc} = m^{Ic} \sum_{\text{edges}} dm^e \tag{154}$$

Thus the mass fluxes computed there are saved and reused during the transport of all other quantities. The edge value of the internal energy is computed by the same scheme as that for the density. In fact, the calculation in Equations 140 through 147 is repeated exactly with the exception of the substitution of e for ρ throughout. Now, if $e^I = e_o$, then, just as for ρ above, $e^e = e^o$. In this case Equation 153 becomes

$$e^F = e_o \{m^{Ic} + \sum_{\text{edges}} dm^e\} / m^{Fc} \tag{155}$$

which, by Equation 154, implies $e^F = e_o$.

8.5.3 Monotonicity of the Transport Scheme--Another significant property for a transport scheme is monotonicity. A scheme is monotonic if the transport of a quantity from one cell to a neighbor, in the absence of any other transport, results in the final densities of that quantity in those cells lying between the initial densities, and in the original order. The mass flux based transport scheme described above is monotonic if donor cell differencing is used, though distinctly non-monotonic if any linear interpolation is used. This will demonstrated for the transport of internal energy. The superscripts + and - be used to designate quantities belonging to the cells to which mass is added and from which mass is

removed, respectively. With this notation, donor cell differencing corresponds to the choice $e^e = e^{H-}$. For the case of mass transport toward higher energy, the monotonicity property is easily stated as

$$e^{H-} \leq e^{H+} \longrightarrow e^{H-} \leq e^{F-} \leq e^{F+} \leq e^{H+} \tag{156}$$

Since there is mass flux across only one edge Equation 155 becomes

$$e^{F-} = \frac{1}{m^{Fc-}} \left(e^{H-} m^{Ic} - e^{e} \left|dm\right|\right) \tag{157}$$

Because

$$e^{H-} \leq e^{e} \leq e^{H+} \tag{158}$$

it is clear that

$$e^{H-} - \left(e^{H+} - e^{H-}\right) \frac{\left|dm\right|}{m^{Fc-}} \leq e^{F-} \leq d^{H-} \tag{159}$$

with equality on the right only if $e^e = e^{H-}$. The remainder of the inequalities required to prove monotonicity for donor cell differencing follow in an analogous fashion. That strict inequality on the left in Equation 158 implies strict inequality on the right of Equation 159 shows that non-donor cell differencing produces non-monotonicity. When a transport scheme is non-monotonic, a propagating front separating two ostensibly uniform states will have excursions above and or below the limit values. Such an excursion is particularly troublesome when it causes a sign change in the energy! It is for this reason that donor cell differencing is used almost exclusively on real problems.

8.5.4 Cylindrical Effects from Convective Derivatives--The last two integrals of Equation 96, having vector integrands, generate terms proportional to $w_c - w^H$. Of course, the coordinate system does not move in the $\hat{\theta}$ direction, i.e., $w_c = 0$, and thus these terms are proportional to w^H. Consider $\bar{v} \cdot \overline{\nabla F}$ in cylindrical symmetry

$$\bar{v} \cdot \overline{\nabla F} = \left(u \frac{\partial}{\partial r} + v \frac{\partial}{\partial z} + \frac{w}{r} \frac{\partial}{\partial \theta}\right) \left(F_r \hat{r} + F_z \hat{z} + F_\theta \hat{\theta}\right)$$

$$= \left(u \frac{\partial}{\partial r} + v \frac{\partial}{\partial z}\right) \bar{F} + \frac{w}{r} \frac{\partial}{\partial \theta} \left(F_r \hat{r} + F_z \hat{z} + F_\theta \hat{\theta}\right) \qquad (160)$$

Since the coordinate functions F_r, F_z, and F_θ are independent of θ by the symmetry assumption, only the basis vectors contribute to the θ derivative term and thus

$$\bar{v} \cdot \overline{\nabla F} = \left(u \frac{\partial}{\partial r} + v \frac{\partial}{\partial z}\right) \bar{F} + \frac{w}{r} \left(F_r \hat{\theta} - F_\theta \hat{r}\right) \qquad (161)$$

For $\bar{F} = \bar{v}$, these terms are responsible for conservation of angular momentum and centripetal acceleration, while for $\bar{F} = \bar{B}$, they account for the turning of trapped field lines due to fluid rotation. They seem more appropriately included with the Lagrangian hydrodynamics, and appear there in explicit form. These terms are omitted from the difference formulas in the convective processes.

8.5.5 Magnetic Flux Transport--The difference form for the fourth of Equations 96, which governs the transport of magnetic field, is thus written

$$\bar{B}^F = \left\{m^{Ic} \bar{B}^H + \sum_{\text{edges}} \bar{B}^e \, dm^e\right\} / m^{cF} \qquad (162)$$

Remember that the cell mass appears here to remove the effect of the divergence of the fluid velocity which is properly included in the difference equations of the Lagrangian hydrodynamics. The edge value $\bar{B}^e$ is given by Equations 140 and 141 with ρ replaced by $\bar{B}$ and $\beta = \beta_v$ of Equation 143.

8.5.6 Momentum Transport--The transport of momentum, described by the third of Equations 96, differs from the other three transport processes described above because the velocities, which play the role of densities, are vertex rather than cell quantities. One possibility is to form and

transport the cell centered momentum using the cell mass and the velocity averaged from the four vertices, and then reform the vertex velocity after the transport step. That is very diffusive, since it transfers momentum across a cell even in the absence of mass flux. Brackbill's approach in MOQUI was to use a separate vertex control volume similar to that of Figure 8, and to transfer momentum on the basis of the mass fluxes between those volumes. Those difference equations suffered from lack of the homogeneity, and hence a grid moving through a uniform velocity field introduced non-uniformity. The problem was caused by the different mass fluxes used for mass and momentum transport.

It is possible to avoid both of these problems at the expense of some increased computational effort and additional complexity. The scheme required is based on the transport of four cell/vertex momenta for each component of the velocity. For each vertex indexed by k = 1,...,4 those momenta are given by

$$\bar{p}_k^{Hc} = m^{Ic}\,\bar{v}^{Hv(k)} \tag{163}$$

The flux step

$$\bar{p}_k^{Fc} = \bar{p}_k^{Hc} + \sum_{\text{edges}} \bar{v}_k^{He}\, dm^e \tag{164}$$

uses exactly the same mass fluxes as for the transport of mass. After the flux step is performed the new vertex mass is computed from

$$m^{Fv} = \frac{1}{4} \sum_{\substack{\text{adjacent}\\ \text{cells}}} m^{Fc} \tag{165}$$

Then the new vertex momentum

$$\bar{p}^{Fv} = \frac{1}{4} \sum_{\substack{\text{adjacent}\\ \text{cells}}} \bar{p}_{k(c)}^{Fc} \tag{166}$$

is obtained, and from it and the vertex mass, the new vertex velocity is obtained by

$$\overline{v}^{Fv} = \overline{p}^{Fv}/m^{vF} \tag{167}$$

In the cylindrical case, the θ velocity is first replaced by

$$(r^F - u_c dt)w^{Hc} \tag{168}$$

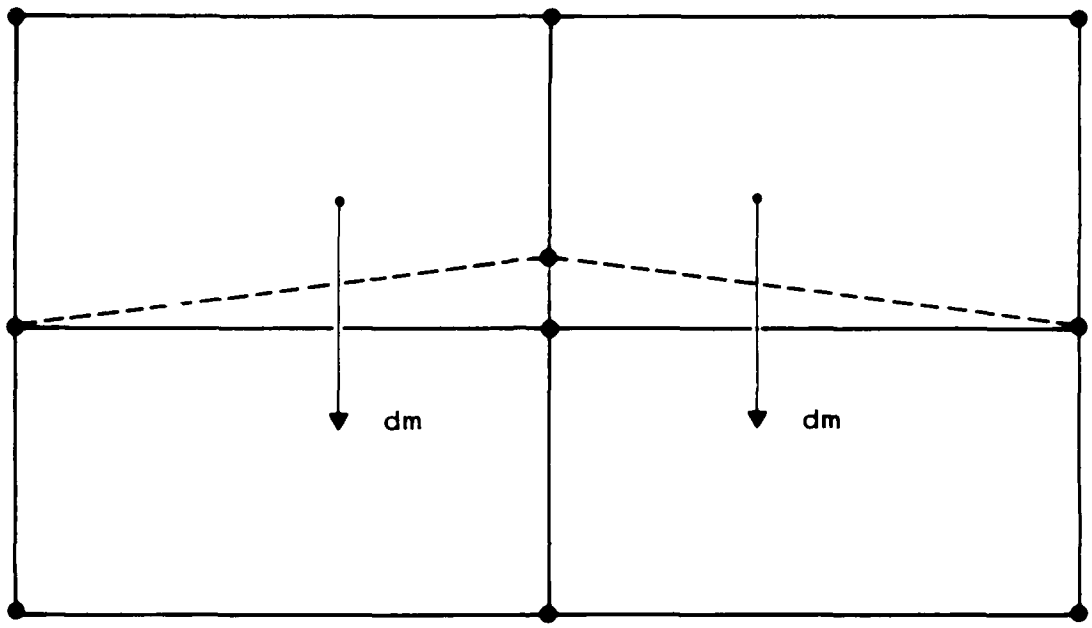
and divided by r^F after Equation 167 is performed.

This scheme can easily be shown to conserve total momentum exactly. The precise statement which can be proven is

$$\sum_{\substack{\text{all}\\\text{vertices}}} \overline{p}^{Fv} = \sum_{\substack{\text{all}\\\text{vertices}}} \overline{p}^{Hv} + \sum_{\substack{\text{exterior}\\\text{edges}}} \frac{1}{4} \sum_{k=1}^{4} \overline{v}_k^{He}\, dm^e \tag{169}$$

That is, the change in total momentum is caused solely by fluxes through the problem boundaries. Furthermore, when there is no mass flux, there is no change in the velocity.

8.5.7 Energy Conservation of the Transport Scheme--The total energy is not conserved by the transport scheme described above. The error occurs because neither magnetic energy nor kinetic energy is conserved exactly. The change in kinetic energy due to the transport scheme can be estimated by considering the transport of mass and momentum due to the relative motion of the grid and fluid at a single vertex. Figure 14 shows a uniform rectangular grid near a central vertex, and a new location for that vertex and its associated edges. The mass fluxes, dm, which this grid motion causes are also shown; the cell mass m is the same in all cells. The velocity $\overline{v}$ is assumed directed vertically downward at all vertices. This



R-874

Figure 14. Configuration for computation of error in kinetic energy due to transport.

velocity is taken to be constant along the horizontal grid lines, but a constant ratio γ is assumed between velocities at vertices connected by a vertical edge. The ratio is taken in such a way that if the velocity decreases in the direction of the mass flux shown in the figure, then $\gamma > 1$. The motion of this single vertex changes the momentum of all 9 vertices shown in the figure. Assuming donor cell averaging, the relative change in total kinetic energy, normalized to 9 times the kinetic energy of the central vertex, is

$$(\gamma - 1)^2 \left(1 + \frac{1}{\gamma^2}\right) \left\{\frac{1}{9}\,\varepsilon + \frac{1}{24}\,\varepsilon^2 + 0\,(\varepsilon^3)\right\} \tag{170}$$

where $\varepsilon = dm/m$. Thus kinetic energy is lost where the grid moves toward higher velocity, and gained where it moves toward lower velocity.

It is possible to force the conservation of total energy by transporting it instead of internal energy, as described by Brackbill in (Ref. 5). This scheme is optional in the code. When it is used, a loss of kinetic energy during the momentum and mass transport steps will result in a decrease in the internal energy. The integrated effect may be large enough to cause the internal energy of some cell to become negative. In addition to being unphysical, this results in interpolation off the edge of the equation of state tables, and sometimes produces negative temperature or negative squared sound speed. As the square roots of these quantities figure in the computation of Spitzer diffusivities and time-step, respectively, this will cause the simulation to terminate with a floating point error.

The principal cause of these problems is that the mass fluxes and edge densities of the other fluxed quantities are selected explicitly. A transport scheme which conserves total energy, momentum, and mass and maintains positivity of internal energy and density probably requires the simultaneous implicit determination of the fluxes of these quantities from the differenced forms of the conservation laws. Such a procedure would be expensive to code and to run.

9. DISCRETIZATION OF BOUNDARY CONDITIONS

Each difference equation detailed in the previous sections applies at any vertex or cell where all data its evaluation requires is defined at all surrounding vertices or cells. Clearly, there must be edges or boundaries since only a finite amount of data may be stored. Thus some data around the edge of the problem region must be generated by separate equations; these are the difference forms of the physical boundary conditions. There are two distinctly different ways they could be applied.

In the first, the physical boundary conditions are applied to determine expressions for the data outside but adjacent to the edge of the problem region in terms of data inside the problem region. Then, different difference equations for the edges are derived by substituting these expressions into the general difference equations.

The second approach avoids the derivation of the boundary difference equations; space is provided for the surrounding data, and those values are computed explicitly from the expressions derived from the physical boundary conditions. Thus the full difference equations may be applied to the boundary cells or vertices, just as to the interior cells and vertices. The extra storage locations are referred to as ghost cells or ghost vertices, depending on the type of the data under consideration, and this approach is called the ghost cell technique. The cells and vertices where the full difference equations are applied are referred to as real cells or real vertices.

In MACH2, boundary conditions throughout the code are applied using ghost cells. The principal advantage of this choice is that the expressions for the exterior data are simpler to code than the boundary difference equations. The principal disadvantage is the extra storage required. A lesser disadvantage, the extra computations required at the boundary, is of negligible effect, since these are carried out in a vectorizable loop.

Since simpler code is more reliable, the choice has the effect of using slightly greater computer resources to reduce the physicist resources required to run a given simulation.

Ghost cell boundary condition application works well with finite volume differencing, because only first order differences are computed. To correctly compute vertex differences of cell centered quantities along the boundary, it is necessary to fill that boundary's ghost cells with appropriate data before the difference computation. However, the cell centered differences of vertex data along a boundary do not reference ghost vertex data; thus it is not usually necessary to fill ghost vertices with data before performing the differences.

9.1 LOCATION OF GHOST VERTICES

The geometric coefficients described in Section 7.4 are the principal exception to the rule just expressed. They are differences of the vertex coordinates, and are thus cell quantities. Their values are required in the ghost cells to compute boundary differences such as the gradient or curl of cell centered quantities, as in Equations 121 and 133. While these ghost cell values could be determined by symmetry considerations, it is more reliable to set ghost vertex locations and compute the ghost cell values of the geometric quantities exactly as for the real cells.

9.1.1 Boundary Ghost Vertex Location--There are two cases for the formation of ghost vertices at a given boundary of a given block. In the first, another block abuts this one across this boundary. The relevant physical boundary condition is continuity of physical data, so the ghost vertex positions in this block are copied from the first interior row of vertices in the adjoining block. Thus, the ghost cells exactly overlay the first row of cells in the neighbor block as in Figure 15. The proper coupling of physical processes is ensured by the simple expedient of copying data from the neighbor block into the ghost cells of the present block.

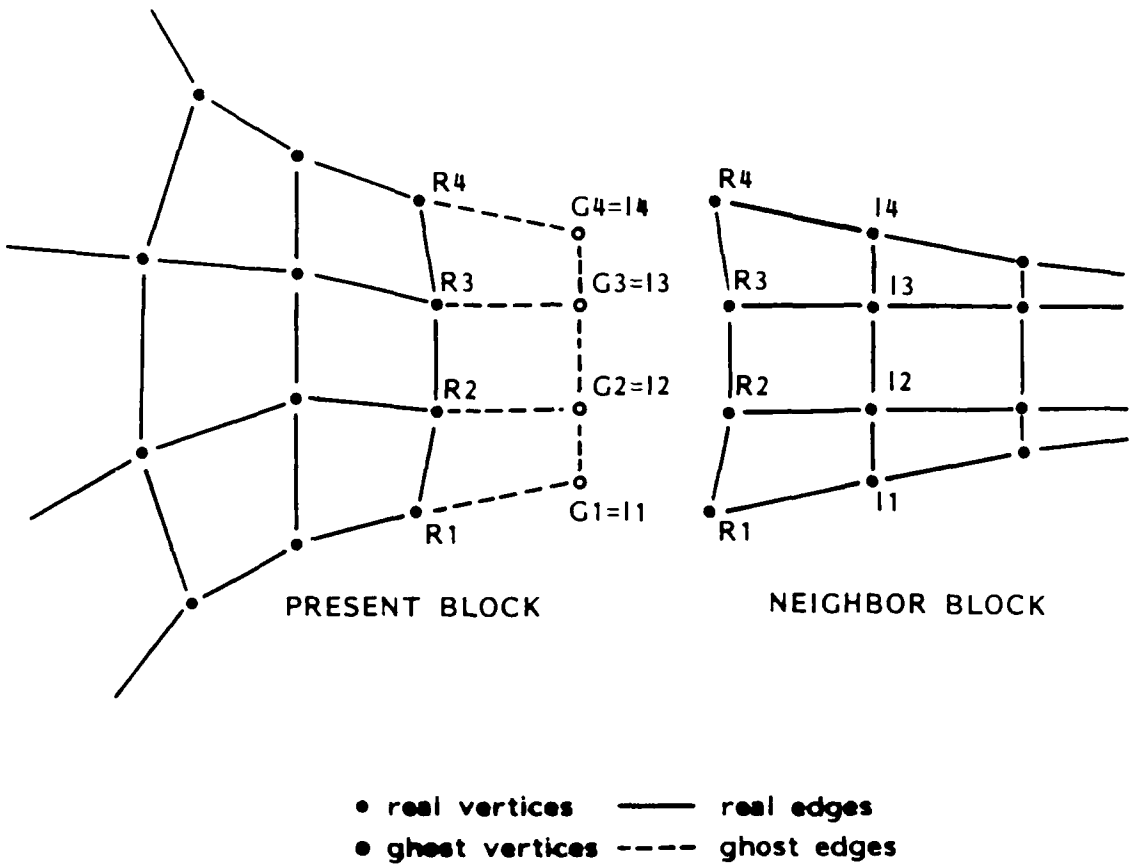


Figure 15. Ghost cells on boundary with neighbor block.

In the second case, i.e., no adjoining block across this boundary, the ghost vertex locations are created using the locations of the vertices in this block. Since most of the physical boundary conditions described in Section 3 involve normal and tangential derivatives, the ghost vertices are positioned to facilitate the computation of those. The position of each vertex in the first interior row is reflected out through the tangent to the boundary at the nearest boundary vertex to form the ghost vertex at that row or column (see Figure 16). The tangent vector used is the centered difference of the adjacent boundary vertices after being rotated outward and normalized to unit length. At the ends of the boundary, the one sided difference is used for the tangent instead of the centered difference.

If the boundary turns too sharply, then this process will generate ghost cells which are bow tied as in Figure 17. In most instances, this will be avoided if the distance from the boundary to the next gridline inward is less than the radius of curvature of the boundary.

9.1.2 Corner Ghost Vertex Location--The scheme described above does not determine the ghost corner vertex location. It is important to position it so that the corner ghost cell formed should, when possible, overlay physically corresponding cells in neighbor blocks, be they ghost or real. There are four cases, corresponding to the number of blocks which may come together at a corner.

If there is only one block incident at the given corner, the corner ghost vertex is positioned so as to make the corner ghost cell a parallelogram as shown in Figure 18a. If the interior angle of the block corner is less than 60 degrees, such as in Figure 18b, the resulting corner ghost cell will have negative area and volume. This restricts the range of problems.

When two blocks meet at a corner, they are required to also meet in a whole edge. Then the ghost corner vertex position is chosen to be the same as the position of the ghost vertex from the neighbor block adjacent

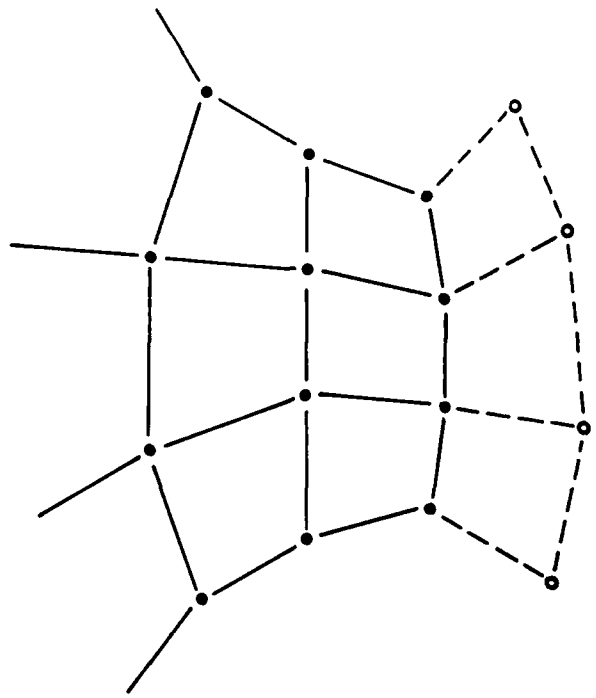


Figure 16. Ghost cells on boundary with no neighbor block.

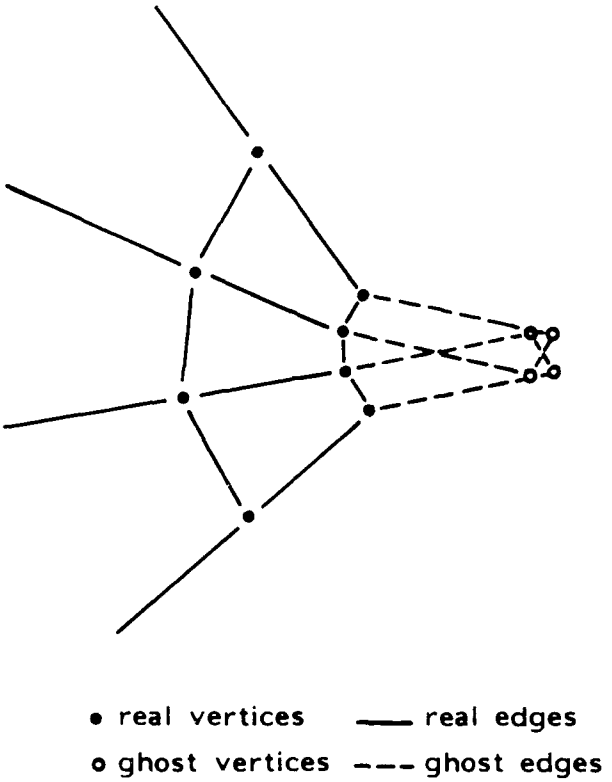
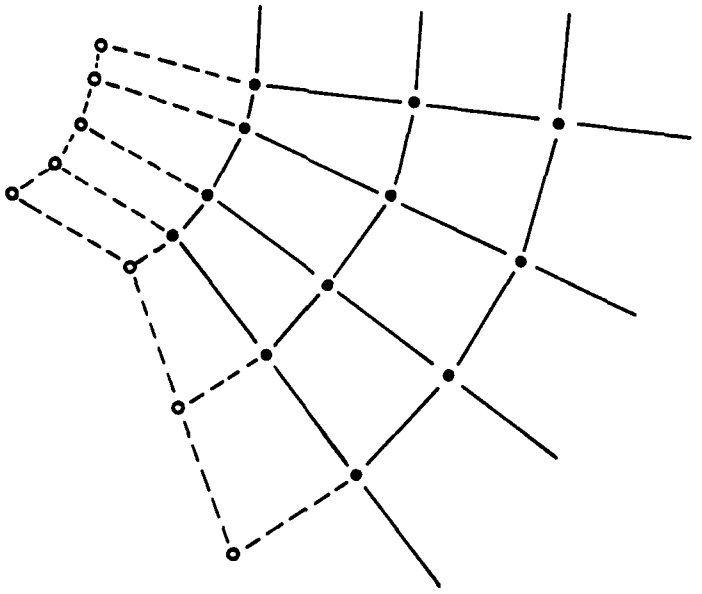


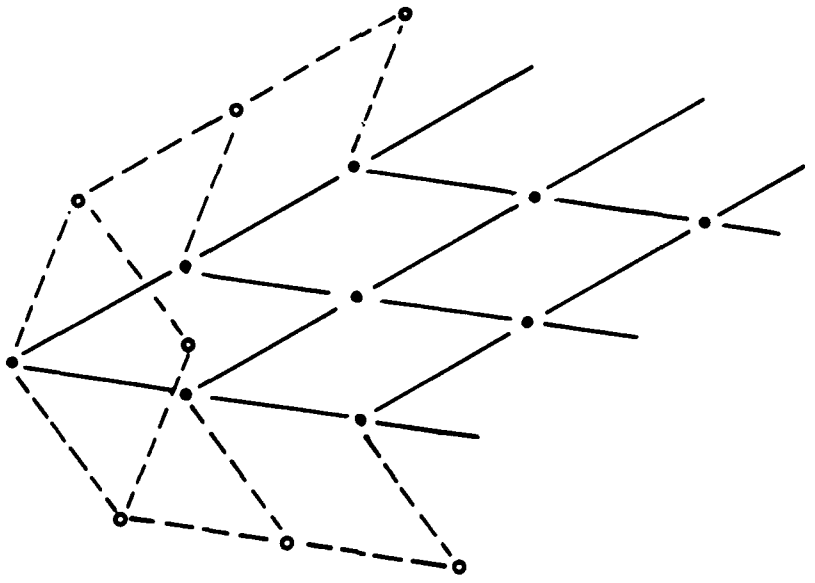
Figure 17. Bow tied ghost cells on curved boundary.



a)

• real vertices —— real edges

o ghost vertices - - - ghost edges



b)

• real vertices —— real edges

• ghost vertices - - - ghost edges

AMAC-R-874

Figure 18. Corner ghost cells at one block corners with interior angle, a) less than 60 degrees, b) greater than 60 degrees.

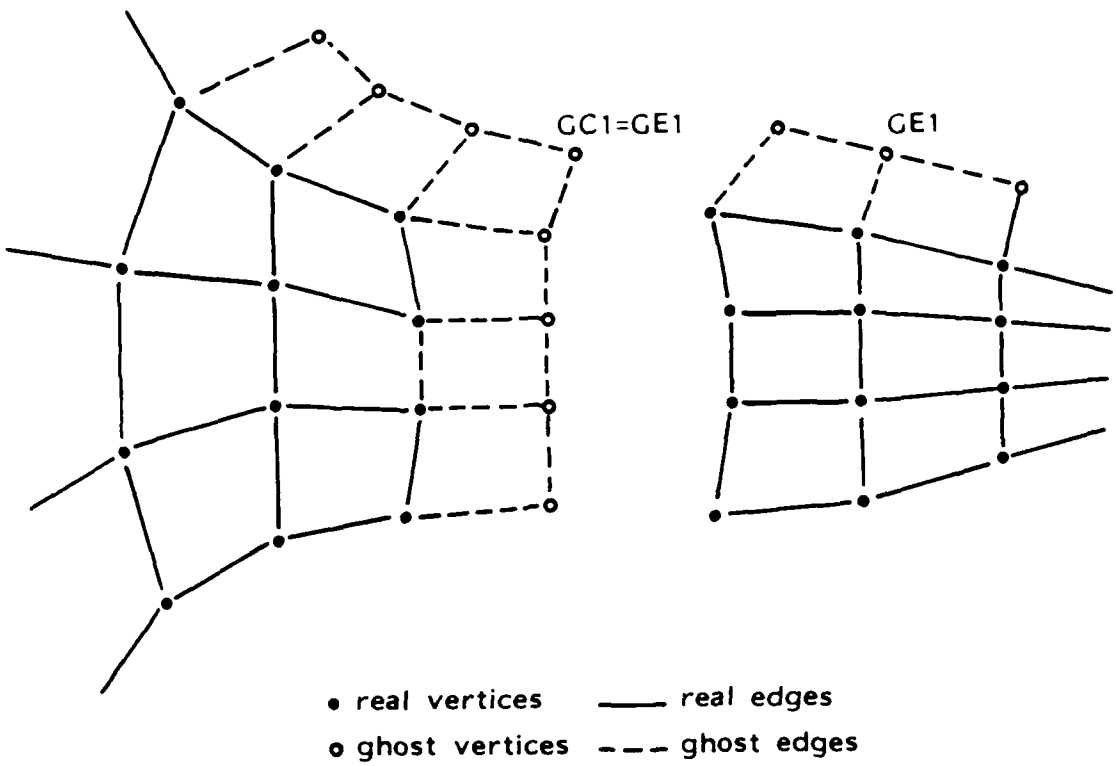


Figure 19. Corner ghost cell at two block corner.

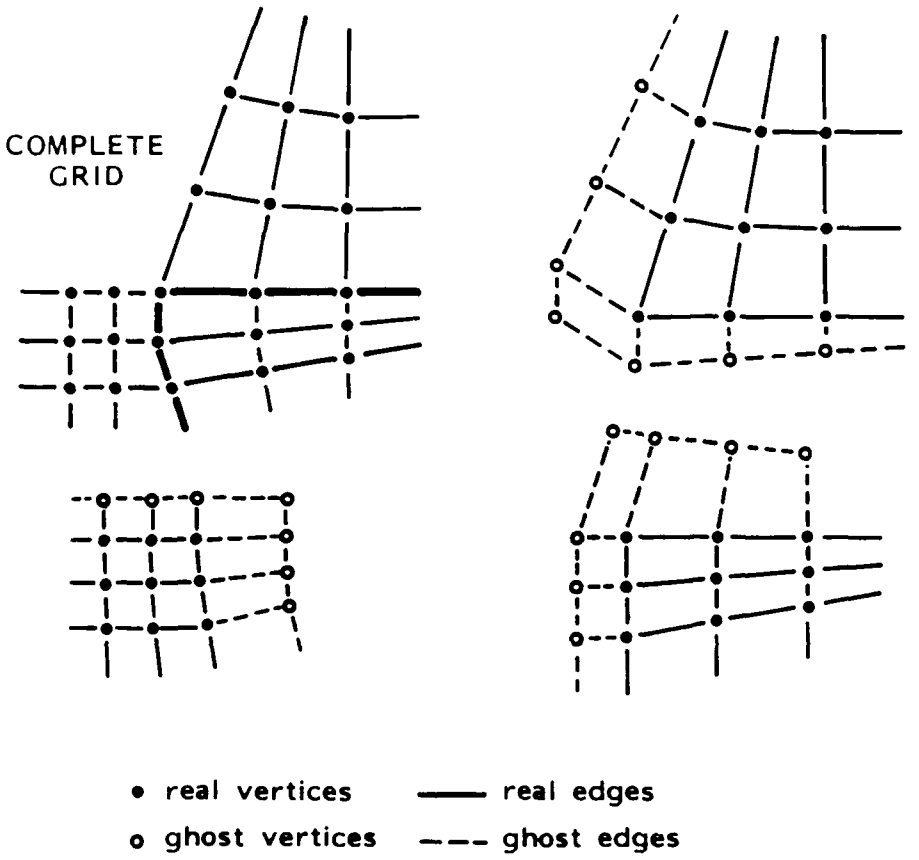
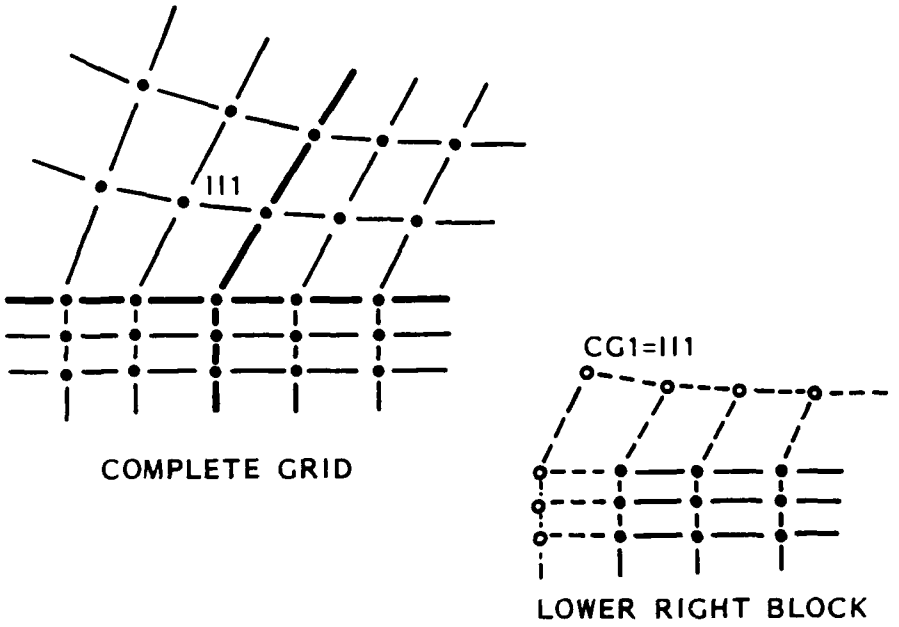


Figure 20. Corner ghost cells at three block corner.



AMRC-R-874

Figure 21. Corner ghost cells at four block corner.

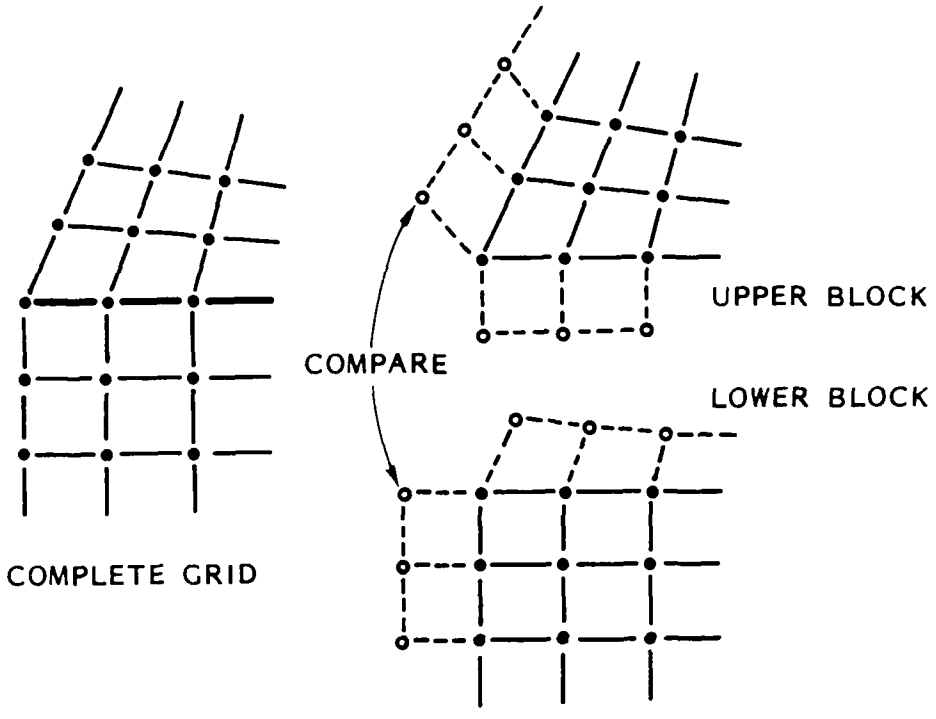


Figure 22. Corner ghost cells at two block corner in curved boundary.

to the real vertex which is itself adjacent to the real corner vertex along the non-coincident edge as in Figure 19. Note that this means that all boundary ghost vertices in all blocks must be positioned before any corner ghost vertex positions are determined.

When three blocks come together at a corner, the effort required to make the ghost vertices overlap near that corner is far too great, even to achieve validity only for a limited set of grids. Hence, the ghost vertex in each block is positioned, just as for the case of a one block corner, by making the ghost corner cell a parallelogram. The four cells adjacent to the corner may thus be quite different from block to block (see Figure 20). This results in different values for the geometric coefficients in cells near the corner, and different values of vertex differences such as current density at the real vertex. For some quantities, these different values may be sensible, as they may be considered to be the different limits of that physical quantity as the corner is approached from different directions. For others, such behavior may be unphysical or may generate numerical instability. In those cases, it may be appropriate to average the different representations together and set all of them to the same value.

The case of four blocks meeting at a corner is easily handled. The ghost corner vertex is placed at the natural position, coincident with the appropriate real vertex in the block diagonally opposite the one under consideration as shown in Figure 21.

When two blocks come together at a corner and the non-coincident edge is not a straight line, a further problem exists with the ghost vertex adjacent to the real corner vertex off the non-coincident edge. Recall that the tangent vector at the end of a boundary points directly toward the next vertex along that boundary in that block. Hence, the tangent vectors at the corner points of the two blocks are not identical. As Figure 22 shows, the two representations of that ghost vertex will then have different positions due the use of different tangent vectors to reflect the interior vertex outward. In this case it is advisable to average these two positions and use the result for both.

Of course, there is an effective radius of curvature at such a corner, and the ghost cells may bow tie there unless care is taken. That radius is approximately

$$\frac{dl + dl'}{2 \sin \theta} \tag{171}$$

where θ is the angular change of the boundary at the corner, and dl and dl' are the distances from the corner to the nearest neighbor gridpoints along the boundary.

9.2 SCALAR BOUNDARY CONDITIONS

The physical boundary conditions described in Section 3 apply to both scalar and vector functions. Most of those applicable to scalar functions are of one of two forms: either

$$\overline{n} \cdot \overline{\nabla}\phi = 0 \quad \text{on } \partial R \tag{172}$$

or

$$\phi = f \quad \text{on } \partial R \tag{173}$$

for some unknown scalar function ϕ and some known scalar function f. In almost all cases, ϕ is a cell centered quantity.

Equation 172 may then be implemented by simply setting the ghost cell values of ϕ equal to its values in the adjacent real boundary cells. Because of the way the ghost cells were constructed above, the normal derivative of ϕ at the boundary will be zero to at least second order in dl. One special case of particular interest is Equation 25, the conducting wall boundary condition for B_θ. This is implemented by forming the cell centered radii in the ghost cell and the real cell, dividing former by the latter, and setting the ghost cell value of B_θ to the product of that ratio and the real cell value of B_θ.

In most cases involving Equation 173, the function f is constant. For cell centered ϕ, it might be argued that the correct way to implement this is to set the ghost cell value of ϕ so that the average of the ghost and real cell values is f, i.e.,

$$\phi_g = 2f - \phi_r \tag{174}$$

where the subscripts r and g stand for real and ghost, respectively. However, as the result of this boundary condition will surely be that

$$\phi_r \sim \phi_g \tag{175}$$

it may be implemented nearly as accurately by simply applying

$$\phi_g = f \tag{176}$$

Once again B_θ is a special case. Equation 26, one form of Ampere's Law, is implemented by setting

$$B_\theta = \sqrt{\frac{\overline{\mu_o}\, I}{2\pi r_c}} \tag{177}$$

where r_c is the cell centered radius and I is the total current flowing between this boundary and the axis of cylindrical symmetry.

9.3 VECTOR BOUNDARY CONDITIONS

The vector boundary conditions of Section 3 are mostly of one of the following three forms

$$\overline{n} \cdot \overline{v} = 0 \text{ on } \partial R \tag{178}$$

or

$$\overline{v} = 0 \text{ on } \partial R \tag{179}$$

or

$$\overline{u}_1 \cdot \nabla (\overline{u}_2 \cdot \overline{V}) = 0 \text{ on } \partial R \tag{180}$$

where each of $\overline{u}_1$ and $\overline{u}_2$ is either $\overline{n}$ or $\overline{t}$. Two such conditions are applied to any vector field on each boundary, with Equation 179 counting as two.

9.3.1 Vertex Centered Fields--If $\overline{V}$ is vertex centered, then Equations 178 and 179 may be applied directly to the real boundary vertex data. Equation 178 is applied by replacing the values of $\overline{V}$ using

$$\overline{v} \leftarrow (\overline{t} \cdot \overline{V})\ t \tag{181}$$

No ghost vertex values need to be set. The tangent $\overline{t}$ is computed using "upwind" differencing for quantities $\overline{V}$ which have direct effect on the grid positioning, such as the grid displacement or fluid velocity. This means that the tangent is computed using a one sided difference in the direction of $\overline{V}$, and amounts to having each gridpoint follow the gridpoint ahead. This is necessary because centered differencing in these cases produces an instability which destroys the smoothness of the grid on a moving boundary. For quantities which are not closely coupled to the grid position, centered differencing for the tangent suffices.

9.3.2 Cell Centered Fields--For cell centered quantities $\overline{V}$, Equations 178 and 180 are the most commonly used. One condition on the normal component and another on the tangential component are required at each boundary. This pair of conditions is combined into a single simple geometric relationship between the ghost cell value and the real edge cell value of the vector field. For example, the perfectly conducting wall condition for B_r and B_z, from Equation 24, is

$$\left.\begin{array}{l} \overline{n} \cdot \overline{B} = 0 \\ \quad \text{and} \\ \overline{n} \cdot \overline{\nabla} (\overline{t} \cdot \overline{B}) = 0 \end{array}\right\} \text{ on } \partial R \qquad (182)$$

These may be accomplished by setting the normal component of (B_r, B_z) in the ghost cell equal to the negative of its value in the adjacent real edge cell, and the tangential component equal to its real edge value.

Geometrically, this is equivalent to reflecting (B_r, B_z) out through the tangent to boundary. It is by this geometric description, tangent reflection, that the operation is identified in the code. Other similar geometric operations which are used include normal reflection, and normal and tangent projection.

9.4 SEQUENCING BOUNDARY CONDITIONS TO CONTROL CORNER VALUES

Which of the two boundary conditions to apply at a corner is too difficult a question to answer a priori. Therefore, this choice is left open so that it may be settled differently in different cases by careful thought or experiment. The choice is made for each physical boundary condition by specifying an order for the boundaries of each block. Since, the boundary conditions are set from one ghost corner to another, the last condition applied on the two boundaries meeting at a given corner determines that corner's ghost cell value.

REFERENCES

1. Jackson, J. D., Classical Electrodynamics, Second Edition (Ch. 10), John Wiley and Sons, Inc. (1975).

2. Cranfill, C. W., EOSPAC: A Subroutine Package for Accessing the Los Alamos EOS Data Library. LA-9278-M, August 1983, Los Alamos National Laboratory, Los Alamos, New Mexico.

3. Holian, K., ed., T-4 Handbook of Material Properties Data Bases, Vol. Ic: Equations of State, LA-10160-MS, November 1984. Los Alamos National Laboratory, Los Alamos, New Mexico.

4. Jackson, J. D., Classical Electrodynamics, Second Edition (Sec. 8.1), John Wiley and Sons, Inc. (1975).

5. Brackbill, J. U., and W. E. Pracht, "An Implicit Almost-Lagrangian Algorithm for Magnetohydrodynamics," Journal of Computational Physics, Vol. 13, p. 455-482 (1973).

6. Hirt, C. W., A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," Journal of Computational Physics, Vol. 14, p. 227-253 (1974).

7. Brackbill, J. U., "Coordinate System Control: Adaptive Meshes," Numerical Grid Generation, Joe F. Thompson, ed., Elsevier Science (1982).

8. Frese, M. H., "A Two-Dimensional Complex Region Mesh Generator," AMRC-R-687, Mission Research Corporation, Albuquerque, NM, 1985.

9. Zalesak, S., "Fully Multidimensional Flux- Corrected Transport Algorithms for Fluids," Journal of Computational Physics, Vol. 31, 3, p. 335-362 (1979).

END

DATE

FILMED

6-88

DTIC